VICTOR: Dataset Copyright Auditing in Video Recognition Systems

Quan Yuan*, Zhikun Zhang*†, Linkang Du‡, Min Chen§,
Mingyang Sun¶, Yunjun Gao*, Shibo He*, Jiming Chen*‖
*Zhejiang University, ‡Xi'an Jiaotong University, §Vrije Universiteit Amsterdam,
¶Peking University, ‖Hangzhou Dianzi University
Email: *{yq21, zhikun, gaoyj, s18he, cjm}@zju.edu.cn, ‡linkangd@gmail.com, §m.chen2@vu.nl, ¶smy@pku.edu.cn

***Abstract*—Video recognition systems are increasingly being deployed in daily life, such as content recommendation and security monitoring. To enhance video recognition development, many institutions have released high-quality public datasets with open-source licenses for training advanced models. At the same time, these datasets are also susceptible to misuse and infringement. Dataset copyright auditing is an effective solution to identify such unauthorized use. However, existing dataset copyright solutions primarily focus on the image domain; the complex nature of video data leaves dataset copyright auditing in the video domain unexplored. Specifically, video data introduces an additional temporal dimension, which poses significant challenges to the effectiveness and stealthiness of existing methods.**

In this paper, we propose VICTOR, the first dataset copyright auditing approach for video recognition systems. We develop a general and stealthy sample modification strategy that enhances the output discrepancy of the target model. By modifying only a small proportion of samples (*e.g.*, 1%), VICTOR amplifies the impact of published modified samples on the prediction behavior of the target models. Then, the difference in the model's behavior for published modified and unpublished original samples can serve as a key basis for dataset auditing. Extensive experiments on multiple models and datasets highlight the superiority of VICTOR. Finally, we show that VICTOR is robust in the presence of several perturbation mechanisms to the training videos or the target models.

I. INTRODUCTION

Video recognition systems [1] have become increasingly vital in real-world applications, including content recommendation on streaming platforms [2], activity recognition in surveillance systems [3], autonomous driving [4] and healthcare monitoring [5], *etc.* To facilitate the research of video recognition, many high-quality datasets are published by the related research institutions such as DeepMind [6], The Allen Institute for AI [7], and Stanford University [8]. However, it should be noted that the video datasets are published with strict open-source licenses to protect the intellectual property of the data owner. Unauthorized commercial deployment would not only easily violate these terms, but also raise serious legal and ethical issues. According to an investigation by Proof News [9], many companies used YouTube video datasets for AI training without permission, which conflicts with YouTube's terms of service. Therefore, it is crucial to audit the dataset copyright in video recognition systems.

Currently, existing dataset auditing research primarily focuses on image [10–12] and audio [13, 14] domains, while few works have yet explored the dataset auditing for video recognition systems. Existing dataset auditing research can be divided into passive auditing and proactive auditing [15, 16]. Passive auditing, primarily built on membership inference [17], infers whether the data of a user is utilized to train the target model [18, 19]. However, this method requires a large number of queries to the model, which is costly. On the other hand, it usually leads to a high false alarm rate, which is impractical. In contrast, proactive auditing injects certain marks such as radioactive data and backdoor into some samples before the dataset is released for subsequent verification [10, 20]. However, most prior studies either require altering data labels, which can degrade the model's performance on normal tasks, or rely on the knowledge of target model, which is often impractical in real-world scenarios. In addition, compared to image data, the additional spatiotemporal dimension of video data and the diversity of various video models pose significant challenges, making most existing methods difficult to implement or ineffective in practice.

Our Proposal. In this work, we present VICTOR (Video recognition Auditor), a practical approach for auditing the dataset copyright by modifying a small portion of the samples in the published dataset. The core idea of VICTOR is to amplify the impact of published modified samples on the prediction behavior of the target model. By evaluating the difference in the outputs of the suspect model for the modified samples and the original samples, VICTOR determines whether the dataset has been misused. If the difference in the output prediction of the original sample and the modified sample is small, it indicates a high likelihood that the suspect model was trained using the published dataset. The design of VICTOR mainly faces the following challenges:

Challenge 1: How to avoid introducing side effects on model training? One prevalent strategy for influencing model behavior is to embed backdoors into the trained model. How-

[†] Zhikun Zhang is the corresponding author.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.240746
www.ndss-symposium.org

ever, such operations can easily cause backdoored samples to produce incorrect predictions, introducing unexpected or exploitable vulnerabilities. Accordingly, VICTOR chooses to retain the original true label of each modified sample throughout the modification process to prevent potential side effects.

Challenge 2: How to amplify the impact of modified samples with low modification costs? On the one hand, the dataset owners usually have limited information about the target model. On the other hand, video data is often subject to operations such as interception and cropping before being processed by the target model. These characteristics significantly hinder the applicability of traditional image dataset auditing techniques to video-based systems. To ensure that the amplification effect remains robust across diverse target models and video preprocessing methods, VICTOR introduces procedural noise [21] into all frames of selected samples. This subtle perturbation effectively enhances the behavioral difference between the released and original samples on the target model, without compromising the visual semantics.

Challenge 3: How to achieve high-precision auditing with low false positive rates? On the one hand, it is challenging to design a reliable mechanism for detecting dataset misuse. On the other hand, if both samples produce low confidence scores for the correct label, their outputs may appear similar, thus obscuring the basis for auditing and increasing the risk of misjudgment. To enable reliable detection of dataset usage, VICTOR employs a subset of published original samples and their corresponding unpublished modifications to estimate a decision threshold. The auditing is carried out via hypothesis testing on the sequence of output differences between unpublished original and published modified samples. Moreover, VICTOR introduces post-processing steps to address scenarios in which both types of sample produce low output probabilities, thereby reducing misleading in the auditing decision.

Evaluation. We conduct experiments on multiple representative video recognition models and three classic open-sourced video datasets to illustrate the effectiveness of VICTOR. The experimental results indicate that VICTOR can effectively audit the dataset usage across various settings. For instance, by modifying only 1% of the samples, VICTOR can achieve up to 100% auditing accuracy across multiple datasets and suspect models. We further analyze the effectiveness of various components and explore the impact of different parameter settings.

Robustness. In practice, the target model might be equipped with various obfuscation techniques to hinder the dataset auditing. Therefore, we conduct the experiments to validate the robustness of our proposed VICTOR. In our work, three representative interference strategies (*i.e.*, input preprocessing, training intervention, post-adjustment) in a general machine learning model pipeline are considered. We observe that the performance of VICTOR only slightly drops, which shows the robustness of VICTOR.

Contributions. In summary, the main contributions of the paper are three-fold:

- To our knowledge, VICTOR is the first dataset copyright auditing approach for video recognition systems.
- We propose a label-invariant perturbation mechanism and a behavior difference-based verification strategy to enable effective auditing. In particular, VICTOR injects procedural noise into a small fraction (*e.g.*, 1%) of the dataset without altering the label. This modification amplifies the influence of altered samples on the target model while effectively keeping visual content and model utility. Combined with careful designs such as threshold estimation and hypothesis testing, VICTOR establishes a reliable foundation for video dataset auditing.
- We conduct comprehensive experiments on multiple models and datasets to illustrate the effectiveness and robustness of VICTOR. VICTOR is open-sourced at https://github.com/sec-priv/VICTOR.

II. RELATED WORK

The dataset copyright auditing methods can be divided into passive auditing and proactive auditing methods, depending on whether the original dataset is modified.

Passive Auditing. The solutions in passive auditing are usually implemented based on the idea of membership inference attack [17, 22–24]. The fundamental principle of membership inference is to identify distinct characteristics between data that has been used in training and data that has not. The existing passive auditing methods can be classified into decision boundary-based and behavior characteristics-based methods [15]. For the decision boundary-based methods [25–29], the core intuition is that samples near the decision boundary in the training dataset are critical for classification. Therefore, by extracting boundary information from the model, the dataset owner can infer whether a particular dataset was used in its training. For the behavior characteristics-based methods [19, 22, 30–34], the dataset owner utilizes the model's outputs and hidden representations as the key basis to finish the auditing. In particular, the model's behavior include the loss value [35], the log-likelihood value [36, 37], and the disparity between the outputs of the target and the shadow models (*i.e.*, models trained on datasets that are similar to the training dataset of the target model) [38–40]. However, these passive auditing approaches usually face significant challenges such as high false positive rates and the need for intensive queries to target models. In light of the low accuracy and poor robustness of passive auditing solutions, we focus on the proactive auditing strategy to safeguard the dataset copyright in this work.

Proactive Auditing. A typical solution in proactive auditing is radioactive data-based auditing [10, 12, 41, 42]. This type of method injects an optimized radioactive mark into the vanilla training images. In practice, the radioactive marks need to be propagated to the image space. If the marked data are used in the training, the classification model is updated with both the features and the radioactive mark. In the copyright validation, the auditor detects the distribution deviation induced by the radioactive marks. However, the distribution shift may be slight by a single marked sample. Thus, the dataset owner needs to inject a large number of marked images into the original dataset to provide statistical evidence that the model is trained on marked images. In addition, recent studies indicate that the performance of radioactive data-based methods is limited [16, 43].

Another classic method of proactive auditing is backdoor-based auditing. In this setting, the dataset owner embeds backdoors (or called triggers) to the original dataset. If a model is training using this modified dataset, the model will perform normally for benign samples. When specific backdoors are present, the model's predictions will change dramatically. Depending on whether the true labels of the samples are altered during the backdoor injection, the backdoor-based method can be classified into dirty-label backdoor [11, 44–48] and clean-label backdoor methods [20, 49, 50]. The dirty-label backdoor methods have better effectiveness, and the clean-label backdoor methods achieve better stealthiness. However, both the two type of methods are prone to introduce harmful influence and potential security risks [51]. The attacker can exploit the backdoor to interfere with the model performance on other normal samples [13].

Recently, Huang *et al.* [16] proposed a data auditing framework and applied it to image classifiers and foundation models. The proposed framework leverages the existing black-box membership inference method, together with a sequential hypothesis testing to implement dataset auditing. However, this approach requires a substantial proportion of data marking in the published version. Chen *et al.* [43] employ a data marking component to mark the target data and adopt a membership inference-based verification process for auditing. However, this approach requires substantial modifications to the original images, which significantly degrades image quality. Moreover, the dynamic nature of video data makes the auditing performance of the above methods insignificant.

III. Preliminaries

A. Video Recognition

Video recognition systems are designed to analyze and interpret the spatio-temporal content of videos, with the goal of assigning semantically meaningful labels to entire clips or localized temporal segments [52]. This system has already been widely applied for various fields, such as action recognition, gesture analysis, healthcare monitoring, and content categorization [53].

Before being fed into the models, raw video inputs typically undergo several pre-processing steps [54]. These include frame extraction, spatial resizing, temporal sampling (*e.g.*, uniform, random, or dense sampling), and modality transformation (*e.g.*, computing optical flow or extracting RGB and depth channels). These pre-processing operations are essential for reducing computational overhead, maintaining temporal consistency, and enhancing the extraction of spatio-temporal features [55].

Over the past decade, researchers have proposed various model architectures for video recognition, which can be categorized into three types: 2D CNN + RNN, 3D-CNN, and Transformer-based models. The 2D CNN + RNN methods utilize 2D CNN for frame feature extraction and RNN for capturing temporal dependencies between them [56–59]. To learn stronger spatial-temporal representations, 3D CNN-based methods were proposed [60–64]. These methods utilize 3D kernels to jointly leverage the spatio-temporal context within a video clip. Inspired by the success in natural language processing, transformer-based models are proposed [65–67]. This type of method adopts self-attention to gauge the relevance of various frames.

B. Threat Model

Application Scenario. This paper aims to address a critical gap in video dataset copyright protection: detecting unauthorized commercial use of open-source licensed datasets. Figure 1 illustrates a typical application scenario where the data owners publish the dataset to the public. In order to protect the dataset copyright, the dataset owner modifies some samples (videos) in the original dataset. The modified samples and other original samples constitute the dataset D_1 and are released to the public. A malicious dataset user (attacker) can download the open-sourced video datasets and then train their models for unauthorized purposes. To confirm whether the trained models use the dataset D_1, the auditor queries these models to obtain the auditing results.

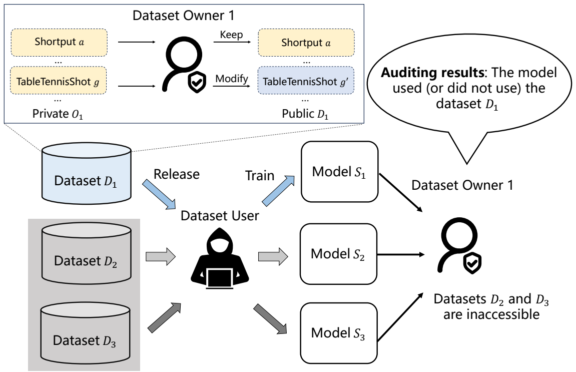

Fig. 1: An example of the application scenario.

Attacker's Capabilities. We assume that the attacker can download the published video dataset and train any video recognition model with the dataset. The attacker may adopt various evasion strategies (such as input preprocessing, training intervention, and post-adjustment) to bypass the auditing mechanisms.

Auditor's Capabilities. The auditor has complete knowledge of the protected dataset and can modify some samples of the dataset prior to release. In addition to its own dataset, we consider that the auditor is unaware of the dataset information from other dataset owners. To mimic the real-world application, we consider that the auditor has black-box access to the suspect model. Note that this is the most general and challenging scenario for the auditor.

IV. Victor

A. Can Image Auditing Methods Apply to Video?

Before introducing our approach, one might wonder why image auditing techniques cannot be applied directly to video. The challenges faced when applying the image auditing methods to video are as follows:

Flexible Video Lengths. The complexity of video data (including additional temporal dimension) is higher than image data. In particular, video data comprises sequences of frames with uncertain lengths, which are typically preprocessed through operations such as frame sampling and cropping before being input into neural networks. Under these conditions, the difference in the amount of information between the raw data and the model input data is significant, making the traditional image auditing methods difficult to implement or ineffective.

Complex Video Models. Due to the increased architectural complexity of video recognition models and their reliance on aggregated information from multiple frames, image auditing techniques designed to perturb the output distribution are generally ineffective especially when the target model architecture and data pre-processing steps are unknown. The greater robustness of video recognition models significantly increases the difficulty of influencing the model output.

Harmful Backdoor Influence. Current image auditing approaches based on backdoor injection typically introduce harmful triggers into the training data. These triggers may not only compromise the model's performance but also introduce exploitable vulnerabilities that attackers can leverage.

B. Motivation

Given that the dataset owner has the ability to modify the data prior to release, our core idea is to amplify the behavioral differences of these modified samples across different target models (*i.e.*, models that have used the dataset and those that have not). During the copyright verification phase, these amplified differences can then be evaluated to determine whether the dataset has been used for training.

For the design goal of dataset auditing, we aim to achieve the following objectives: 1) Modify only a small fraction of the dataset at a low cost to reduce the discrepancies between the published and original datasets; 2) Ensure that the proposed method can reliably identify whether the released dataset was misused by leveraging the behavioral differences in the suspect model across various data pre-processing procedures and model architectures; 3) Avoid introducing harmful backdoor influence that could degrade the performance of the target model; 4) Maintain robustness against various adaptive attacks.

C. Overview

As depicted in Figure 2, the workflow of Victor mainly consists of three phases: Sample modification, sample selection, and copyright verification. If the suspect model is trained on the target dataset, the auditor should output a positive auditing result; otherwise, a negative auditing result. We also summarize the frequently used mathematical notations in Table I.

TABLE I: Summary of mathematical notations.

Notation	Description
O	Original dataset
Q	Modified dataset
D	Published dataset
U	Unpublished dataset
ε	Perturbation budget
δ_{per}	Parameter set for Perlin noise
M	Evaluation model
S	Suspect model
P	Posterior probability
$\mathbb{R}$	Reference set
$\mathbb{M}$	Modification set
$\mathbb{E}$	Remaining set
r_c	Ratio of candidate set
r_m	Ratio of modification set
r_r	Ratio of reference set
H	Upper limit of the difference threshold
B	Boundary of low probability
β	Boundary coefficient
α	Significant level

Phase 1: Sample Modification. In this phase, an evaluation model is first trained based on the original video dataset. In addition, we can generate the modified version for each sample (video) in the original dataset. Then, the prediction outputs of original and modified versions can be calculated by the evaluation model. These predictions can be served as the selection basis in the following phase. The details of Phase 1 are in Section V-A.

Phase 2: Sample Selection. Based on the output results from Phase 1, we hope to find the samples that are most likely to achieve amplification effect as the final published samples. Here, we choose to calculate the difference in the predictions of the evaluation model on the original and modified datasets. The samples with larger differences form the candidate set, while those samples with smaller differences construct the remaining set. Next, we select a subset of candidate samples as the modification set and another subset as the reference set. Further, the other samples in the candidate set that were not selected are also included in the remaining set. We then release the public dataset, which includes the modified samples of the modification set, and the original samples of the reference and remaining sets. The details of Phase 2 can be found in Section V-B.

Phase 3: Copyright Verification. During this phase, Victor determines whether the dataset has been misused by leveraging the samples from the modification and reference sets. The key insight is to assess whether there exists a statistically significant difference in the suspect model's performance between the published and unpublished samples from the reference and modification sets. The details of Phase 3 are referred to Section V-C.

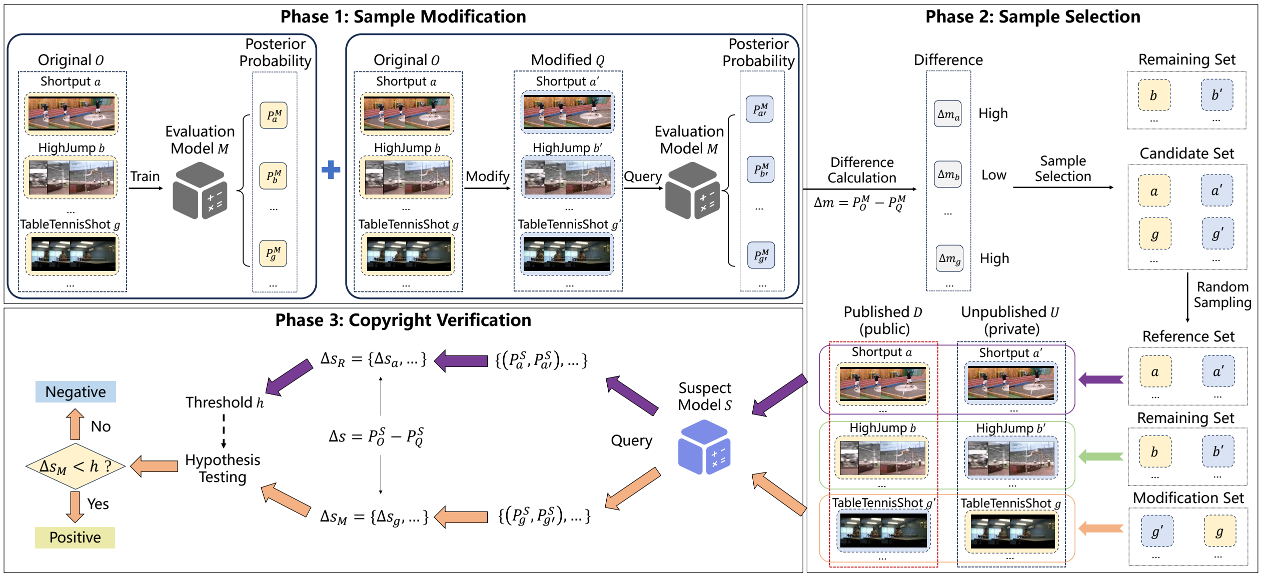

Fig. 2: VICTOR overview. VICTOR is composed of three phases: Sample modification, sample selection, and copyright verification. In the sample modification phase, an evaluation model is trained based on the original dataset. In addition, the modified version of original dataset is generated in this phase. In the sample selection phase, VICTOR quantifies the difference between each modified sample and its original counterpart by analyzing the evaluation model's outputs. This difference serves as a key criterion for selection. Samples with larger differences form the candidate set, while those with smaller differences are placed in the remaining set. VICTOR then randomly selects a subset of candidate samples as the modification set and another subset as the reference set. VICTOR proceeds to publish the modified samples from the modification set, along with the original samples from the reference and remaining sets. In the copyright verification phase, VICTOR assesses whether the dataset has been misused by analyzing the suspect model's predictions on the reference and modification sets.

V. Design Details

A. Sample Modification

During the sample modification phase, VICTOR aims to obtain an evaluation model and a modified dataset based on the original dataset. Algorithm 1 illustrates the basic process of this phase.

First, VICTOR trains an evaluation model M based on the original dataset O. Then, the posterior probability prediction of each sample in O can be computed according to the model M. The evaluation model M can serve as a basis for guiding sample selection. Intuitively, a larger prediction discrepancy between the original and modified samples on M indicates a higher potential for amplification, making such samples more suitable for effective auditing.

In addition, VICTOR tries to generate a modified version for each sample in the original dataset O. Our objective is to generate modified samples that remain visually similar to the original videos while keeping the label unchanged and inducing noticeable differences in the output of the target model, under constrained modification costs. However, achieving this goal on video data presents several challenges. First, although a video consists of many frames, only a subset is typically used for model prediction, and both the number and indices of these selected frames are often unknown. Second, video recognition models aggregate temporal information across multiple frames during inference, making them inherently more robust than image models. Furthermore, the architectures of target models are highly diverse, exacerbated by the difficulty of designing universally effective modifications. These factors significantly limit the applicability of traditional image domain solutions in the video domain.

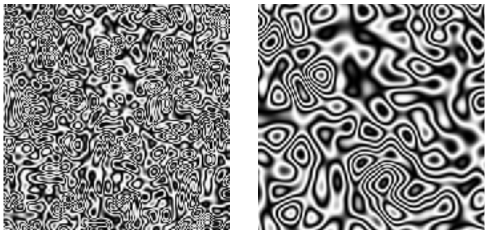

Fig. 3: The example of Perlin noise.

To address the above challenges, we draw inspiration from two-dimensional procedural noise [68], especially Perlin noise [69], which has been shown to effectively reduce prediction confidence in various image classification models [21, 43]. Due to the great versatility and low implementation cost of procedural noise, we choose to inject it into the original video. In particular, procedural noise refers to algorithmically generated patterns that simulate the randomness and irregularity found in natural phenomena. Figure 3 illustrates the examples of procedural noise.

In this work, we extend the classical two-dimensional Perlin noise construction to the spatio-temporal domain to generate

```
Algorithm 1: Sample Modification
Input: Original dataset $O$, perturbation budget $\varepsilon$, the parameter set for three-dimensional Perlin noise $\delta_{\text{per}} = \{\lambda_x, \lambda_y, \lambda_t, \phi_{\text{sine}}, \Omega\}$
Output: The evaluation model $M$, the modified dataset $Q$, the posterior probability vectors $P_O^M$ and $P_Q^M$
// Model training
Train an evaluation model $M$ based on the original dataset $O$
Calculate the posterior probability $P_O^M$ of the samples in the original dataset $O$ by evaluation model $M$
// Noise injection
for each sample $o$ in $O$ do
    Generate the normalized three-dimensional Perlin noise with the parameter set $\delta_{\text{per}}$
    Scale the noise based on the perturbation budget $\varepsilon$
    Add the noise to the original sample and normalize it to $[0, 255]$ to obtain the modified sample $o'$, then add $o'$ to $Q$
    Calculate the posterior probability $P_{o'}^M$ of the modified sample and update it to $P_Q^M$
```

```
Algorithm 2: Sample Selection
Input: Original dataset $O$, the modified dataset $Q$, the posterior probability vectors $P_O^M$ and $P_Q^M$, the ratio of candidate set $r_c$, the ratio of reference set $r_r$, the ratio of modification set $r_m$
Output: The published dataset $D$, the unpublished dataset $U$, the reference set $\mathbb{R}$, the modification set $\mathbb{M}$, the remaining set $\mathbb{E}$
// Difference calculation
$\Delta m \leftarrow P_O^M - P_Q^M$
Sort the values in $\Delta m$
Select samples with values in the first $r_c$ proportion in $\Delta m$ as the candidate set $\mathbb{C}$, and the remaining $(1 - r_c) \cdot |O|$ samples as the remaining set $\mathbb{E}$
// Random sampling
Randomly select $r_r \cdot |O|$ samples from the candidate set $\mathbb{C}$ as the reference set $\mathbb{R}$, and select $r_m \cdot |O|$ samples as the modification set $\mathbb{M}$, where $\mathbb{R}$ and $\mathbb{M}$ do not overlap
Add other $(r_c - r_m - r_r) \cdot |O|$ samples in the candidate set $\mathbb{C}$ that do not belong to $\mathbb{R}$ and $\mathbb{M}$ to the remaining set $\mathbb{E}$
// Dataset division
The published dataset $D$ consists of modified samples of the modification set, as well as original samples of the remaining set and the reference set
The unpublished dataset $U$ consists of original samples of the modification set, as well as modified samples of the remaining set and the reference set
```

dynamic, temporally-coherent perturbations suitable for video data. In addition to the spatial wavelengths λ_x, λ_y, and the number of octaves Ω, we further introduce a temporal frequency control parameter λ_t to regulate variation across the temporal axis.

Let (x, y, t) be a query point in normalized coordinates, and let (i, j, k) be the corresponding corner lattice points, *i.e.*, $i = \lfloor x \rfloor, j = \lfloor y \rfloor, k = \lfloor t \rfloor$. Each lattice corner is assigned a pseudo-random unit gradient vector:

$$\mathbf{g}_{ijk} \sim \mathcal{N}(0, \mathbf{I}_3),$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$ denotes a 3-dimensional normal distribution with zero mean and identity covariance matrix, *i.e.*, $\mathbf{g}_{ijk}$ is sampled independently from a standard multivariate Gaussian. The relative offset from the lattice corner is:

$$u = x - i, \quad v = y - j, \quad w = t - k.$$

We compute the dot product between the gradient vector and the relative offset vector at each of the 8 corners of the unit cube, followed by trilinear interpolation using Perlin's fade function:

$$f(s) = 6s^5 - 15s^4 + 10s^3. \tag{1}$$

Let $P_n(x, y, t)$ denote the interpolated Perlin noise value. We then apply a fractal summation over Ω octaves:

$$S_{\text{per}}(x, y, t) = \sum_{n=1}^{\Omega} \frac{1}{2^n} P_n\left(\frac{2^{n-1}x}{\lambda_x}, \frac{2^{n-1}y}{\lambda_y}, \frac{2^{n-1}t}{\lambda_t}\right). \tag{2}$$

To enhance the perturbation diversity and control high-frequency content, we apply a sinusoidal transformation:

$$G_{\text{per}}(x, y, t) = \sin\left(2\pi \cdot \phi_{\text{sine}} \cdot S_{\text{per}}(x, y, t)\right). \tag{3}$$

Here, ϕ_{sine} is a tunable sine frequency parameter that controls the intensity and granularity of the generated texture. Therefore, the full parameter set for three-dimensional Perlin noise is $\delta_{\text{per}} = \{\lambda_x, \lambda_y, \lambda_t, \phi_{\text{sine}}, \Omega\}$.

The final noise field is normalized to the range $[0, 1]$. Then, we utilize the budget ε as the ℓ_∞-norm constraint on the perturbation. Furthermore, we inject the scaled noise into all frames of the original video and obtain the modified sample, with the constraint of $[0, 255]$. On this basis, we can obtain the posterior predictions of the modified samples. In addition, we clarify that 3D Perlin noise can be replaced with other types of noise in practical deployment (*e.g.*, Gabor noise), which demonstrates the generalizability and stealthiness of VICTOR.

Through the above steps, in the first phase, we can obtain an evaluation model M by training and generate a modified dataset Q, which can be used for subsequent selection.

B. Sample Selection

Based on the modified samples obtained in the first phase, the published modified samples can be selected in this phase. Algorithm 2 provides the process of the sample selection.

First, we compute the prediction difference Δm between the original and modified datasets using the outputs of the evaluation model M. The values in Δm are then sorted, and samples with larger differences are preferred for subsequent auditing. This strategy is based on the intuition that a larger prediction gap indicates a more significant behavioral difference between a model trained on the dataset and one that is not, providing a reliable basis for dataset auditing. Here, we choose to select samples with values in the first r_c proportion in Δm as the candidate set $\mathbb{C}$, the remaining samples are divided into the remaining set $\mathbb{E}$.

Next, we randomly select $r_r \cdot |O|$ samples from the candidate set $\mathbb{C}$ as the reference set $\mathbb{R}$, and select another $r_m \cdot |O|$ samples as the modification set $\mathbb{M}$. The reference and modification sets will be utilized for the auditing in the third phase. Then, the other $(r_c - r_m - r_r) \cdot |O|$ samples in the candidate set $\mathbb{C}$ are assigned to the remaining set $\mathbb{E}$.

According to the original dataset O and modified dataset Q, along with the modification, reference, and remaining sets, we construct two distinct datasets. The first is the public dataset D, which is released and contains the modified samples from the modification set, as well as the original samples from the reference set and the remaining set. The second is the private dataset U, which is kept unpublished and includes the original versions of the modification set, along with the original samples from the reference and remaining sets.

C. Copyright Verification

In this phase, we need to audit whether the published dataset was misused. Considering the complexity and diversity of target models and data processing pipelines, we assess dataset usage by evaluating the performance differences between the reference set and the modification set on the target model. The detailed process is shown in Algorithm 3.

First, for each sample pair (o, o') consist of the original sample o and modified sample o' in the reference set $\mathbb{R}$, we can obtain their predictions on the suspect model S, *i.e.*, P_o^S and $P_{o'}^S$. Then, the prediction difference Δs_o between o and o' can be calculated. After traversing and querying the samples in $\mathbb{R}$, we can get a sequence of difference values, *i.e.*, Δs_R. The purpose of these steps is to obtain a threshold h.

The core design intuition is that if the suspect model S was not trained on the published dataset D, its behavior on the reference set and the modification set should be similar. Here, we define model behavior as the prediction difference between the original dataset O and the modified dataset Q. Conversely, if the suspect model was trained on D, we expect a noticeable behavioral discrepancy between the reference and modification sets. Specifically, the prediction differences for the modification set are desired to be significantly smaller than those for the reference set. This is because the published dataset D includes the modified samples from the modification set and the original samples from the reference set.

To make this distinction operational, we compute the mean prediction difference $\bar{h}$ across all sample pairs in the reference set and adopt it as the decision threshold. However, we observe that the prediction differences within the reference set may exhibit particularly high values, leading to an overly large threshold. Consequently, a higher risk of false positives is prone to occur. To address this, we impose an upper limit H on the difference threshold to improve the reliability of auditing and mitigate misclassification. Regarding the value of upper limit H, we provide a detailed analysis in Appendix A.

Algorithm 3: Copyright Verification

```
Input: The published dataset $D$, the unpublished
       dataset $U$, the modification set $\mathbb{M}$, the
       reference set $\mathbb{R}$, the suspect model $S$, the upper
       limit of the difference threshold $H$, the
       boundary of low probability $B$, the boundary
       coefficient $\beta$, significant level $\alpha$
Output: Auditing result
Initialized $\Delta s_R = \{\}, \Delta s_M = \{\}$
// Model query
for each sample pair (o, o') in the reference set do
    Calculate the output probability of $o$ and $o'$ on the
      model $S$, i.e., $P_o^S$ and $P_{o'}^S$
    Add $\Delta s_o = P_o^S - P_{o'}^S$ to $\Delta s_R$
// Threshold estimation
$\bar{h} \leftarrow mean(\Delta s_R)$
// Range constraint
$h \leftarrow clip(\bar{h}, -H, H)$
// Model query
for each sample pair (g', g) in modification set do
    Calculate the output probability of $g'$ and $g$ on the
      model $S$, i.e., $P_{g'}^S$ and $P_g^S$
    $\Delta s_g = P_g^S - P_{g'}^S$
    // Post processing
    if $P_{g'}^S < B$ and $P_g^S < B$ then
        $\Delta s_g = (1 + \beta)\bar{h}$
    Add $\Delta s_g$ to $\Delta s_M$
Hypothesis testing based on difference sequence $\Delta s_M$
  and threshold $h$ with significant level $\alpha$
```

After computing the threshold h using the reference set, we query the suspect model with the sample pairs (*e.g.*, g' and g) from the modified set and obtain their corresponding prediction differences, denoted as Δs_g. We then apply a post-processing step to refine Δs_g. Specifically, if the predicted probabilities for both the original and modified samples are very low (*i.e.*, below the predefined threshold $B = 1/n_c$, where n_c is the number of output categories), we adjust Δs_g to $(1+\beta)\bar{h}$, where β is a small positive constant (*e.g.*, 0.01). Here, β is utilized to guide the direction of subsequent hypothesis testing and avoid zero-difference exclusion.

The rationale behind this adjustment is: When both predicted probabilities are negligible, the suspect model S is likely not to utilize the dataset D. However, due to the low absolute values, the resulting difference Δs_g may also be small, even smaller than the threshold h. This can mislead the auditor to determine that the dataset D was used. By adjusting Δs_g slightly above the average difference in the reference set, we can effectively mitigate the risk of such false positives and improve the robustness of auditing.

Finally, we employ hypothesis testing to verify whether the suspect model S exhibits dataset misuse. The goal of the hypothesis test is to verify whether the difference in the probability of sample pairs in the candidate set Δs_M is significantly lower than a specific threshold h (calculated from the difference in the probability of sample pairs in the reference set, Line 9 in Algorithm 3). If the probability difference is indeed significantly lower than the threshold (*i.e.*, $\Delta s_M < h$), it means that the suspect model is likely trained using the protected dataset (*i.e.*, the dataset is misused). Conversely, if the probability difference is not significantly lower than the threshold, it determines that the dataset is not misused. Specifically, we adopt the Wilcoxon Signed-Rank Test [70], a non-parametric statistical method designed for paired samples, to implement the hypothesis testing. Unlike parametric tests, the Wilcoxon Signed-Rank Test does not rely on the assumption of normality, making it well-suited for scenarios with small sample sizes or unknown data distributions. Moreover, this method effectively leverages both the sign and rank information of the sample differences, offering higher statistical efficiency and better interpretability. This makes it particularly appropriate for detecting subtle yet consistent behavioral deviations in the model. The null hypothesis H_0 and alternative hypothesis H_1 are as follows:

$$\begin{aligned} H_0 &: \Delta s_M > h, \quad \text{(no significant misuse detected)} \\ H_1 &: \Delta s_M < h, \quad \text{(evidence of dataset misuse).} \end{aligned}$$

The testing procedure is described below:

1) For each paired observation, compute the signed difference relative to the reference threshold:
$$d_i = \Delta s_M^{(i)} - h, \quad i = 1, 2, \ldots, n,$$
where $\Delta s_M^{(i)}$ denotes the i-th observed difference.
2) Exclude all instances where $d_i = 0$ (*i.e.*, no difference).
3) For the remaining non-zero differences, compute the absolute values $|d_i|$ and assign ranks R_i based on these absolute values (average ranks in case of ties).
4) Assign a sign to each rank based on the sign of d_i, and compute the sum of negative ranks:
$$W = \sum_{i:d_i<0} R_i.$$
The statistic W represents the total rank magnitude supporting H_1 (*i.e.*, cases where $\Delta s_M^{(i)} < h$).
5) Use the statistic W to compute a one-sided p-value. If $p < \alpha$ (*e.g.*, $\alpha = 0.01$), reject the null hypothesis H_0 and conclude that the dataset D was misused.

D. Putting Things Together

The aforementioned three phases form the overall process of VICTOR. Algorithm 4 describes the overall workflow of VICTOR. In the first phase, we can train an evaluation model M and generate a modified dataset Q based on the original dataset O, perturbation budget ε and noise parameters δ_{per}. Then, in the second phase, we can select the suitable samples that are likely to achieve the amplification effect according to the posterior probability of samples in O and Q on the evaluation model M. The selected modified samples and other original samples are released as the public dataset D. In the copyright verification phase, we calculate the suspect model's behavior difference on the reference set and the modification set, and apply hypothesis testing to verify whether the published dataset was used.

```
Algorithm 4: VICTOR
Input: Original dataset $O$, perturbation budget $\varepsilon$, the
       parameter set for three-dimensional Perlin
       noise $\delta_{\text{per}} = \{\lambda_x, \lambda_y, \lambda_t, \phi_{\text{sine}}, \Omega\}$, the ratio of
       candidate set $r_c$, the ratio of modification set
       $r_m$, the ratio of reference set $r_r$, the suspect
       model $S$, the upper limit of the difference
       threshold $H$, the boundary of low probability
       $B$, boundary coefficient $\beta$, significant level $\alpha$
Output: Audit result
// Phase 1: Sample modification
$M, Q, P_O^M, P_Q^M \leftarrow$ Algorithm 1$(O, \varepsilon, \delta_{\text{per}})$
// Phase 2: Sample selection
$D, U, \mathbb{R}, \mathbb{M}, \mathbb{E} \leftarrow$ Algorithm 2$(O, Q, P^M, r)$
// Phase 3: Copyright verification
Audit result $\leftarrow$Algorithm 3$(D, U, \mathbb{R}, \mathbb{M}, H, B, \beta, \alpha)$
```

Note that VICTOR can be extended to Top-K or Label-only scenarios. Specifically, different weights are assigned to K publication categories with different rankings to approximate the probability (*e.g.*, the weight is the inverse of the ranking). Then, our current scheme can be directly applied for auditing.

VI. EVALUATION

In this section, we first describe the experimental setup in Section VI-A, and evaluate the overall auditing performance in Section VI-B. Then, we perform ablation experiments to explore the influence of various components of VICTOR in Section VI-C. Moreover, we explore the auditing performance of various methods under Top-K and Label-only settings in Section VI-D. Next, we verify the impact of the parameter setting of VICTOR in Section VI-E. Furthermore, we explore the robustness of VICTOR in Section VI-F. Due to space constraints, we defer the auditing results on the SSv2 dataset to Appendix D. We further include more evaluation results (*e.g.*, the robustness evaluation under common perturbations, auditing performance against adaptive attackers, efficiency analysis) in Appendix of [71].

A. Experimental Setup

Datasets. We evaluate the performance of different methods on three standard benchmark datasets used: HMDB-51 [72], UCF-101 [73], and SSv2 [74]. The first dataset contains 51 categories and a total of nearly $7,000$ videos, and the second dataset consists of more than $13,000$ videos in 101 categories. The last dataset contains more than $160,000$ videos, which consists of 174 categories.

These datasets all belong to action recognition, which is a representative task and widely considered in existing video recognition research. The action recognition requires modeling both spatial and temporal information in videos, which allows us to effectively evaluate the performance of the proposed method. The core idea of VICTOR is to determine auditing decisions based on behavioral differences of the suspect model in different samples. This method design has the potential for generalization since the model's behavioral differences can be characterized in other tasks.

Suspect Models. We verify the effectiveness of our method on four typical models, *i.e.*, I3D [64], SlowFast [61], TSM [56], and TimeSformer [66]. These models are also widely adopted in existing video studies [48, 75].

The I3D and TSM models utilize ResNet-50 pre-trained on the ImageNet dataset to initialize their backbones, and the TimeSformer model adopts the pretrained ViT model to initialize the backbone. The SlowFast model employs a randomly initialized ResNet-50 backbone without pre-training. In this work, we utilize open-sourced MMAction2 [76] to implement the above video recognition models.

Baselines. To our knowledge, there are no other dataset auditing designed for video recognition. Here, we apply the SOTA auditing methods (*i.e.*, ML-DA [16] and MT [43]) for image data to the video domain as the baselines. A detailed description and discussion can be found in Appendix B.

Metrics. We adopt the following metrics to evaluate the auditing performance: Δacc, true positive rate (TPR), F1 score and false positive rate (FPR). Δacc represents the difference between the test accuracy of the model trained using the modified dataset and the test accuracy of the model using the original dataset. TPR measures the proportion of correctly identified positive samples (*i.e.*, trained on the target dataset) among all actual positives. F1 score balances precision and recall, providing a single harmonic mean that is particularly informative under class imbalance. FPR quantifies the proportion of negative samples (*i.e.*, trained on other datasets) incorrectly classified as positive. For FPR, lower is better. For the other three metrics, higher is better.

Experimental Settings. For VICTOR, we set the perturbation budget $\varepsilon = 10$ (*i.e.*, ℓ_∞-norm constraint), the ratio of modification samples $r_m = 1\%$, the upper limit of the difference threshold $H = 0.05$, and the significant level $\alpha = 0.01$. We utilize the I3D model as the evaluation model by default. For the parameter set of Perlin noise, we set $\lambda_x = \lambda_y = 32$, $\lambda_t = 6.4$, $\phi_{sine} = 1$, and $\Omega = 2$. The experiments are conducted with a server with 64-core EPYC AMD CPU and four NVIDIA A6000 GPUs.

Setup. We divide the dataset into two subsets: One designated as the protected dataset (*i.e.*, positive), and the other as the unprotected dataset (*i.e.*, negative). For the protected subset, we generate ten different published versions (*i.e.*, $D_1, ..., D_{10}$) and train ten corresponding suspect models (*i.e.*, $S_{p1}, ..., S_{p10}$) for verification. For the unprotected subset, we first create ten different datasets through random sampling and train ten models (*i.e.*, $S_{n1}, ..., S_{n10}$).

TABLE II: SSIM comparison on various datasets.

Dataset	ML-DA	MT	VICTOR
HMDB-51	0.806	0.681	**0.813**
UCF-101	0.785	0.680	**0.798**

During the verification phase, each published dataset D_i ($i \in \{1, ..., 10\}$) is used to audit its corresponding suspect model S_{pi}, as well as the ten models trained on unprotected data S_{n1} through S_{n10}. In total, across D_1 to D_{10}, we obtain 10 positive samples and 100 negative samples for evaluation.

B. Overall Auditing Performance

In the section, we explore the overall auditing performance of VICTOR and the baseline. First, we provide an illustration of generated videos of various methods, as shown in Figure 4. We further provide the average SSIM comparison of different methods on various datasets, as illustrated in Table II. We observe that the changes introduced by VICTOR are relatively smooth across video frames, whereas ML-DA exhibits more abrupt transitions between frames due to the independent nature of its perturbations. MT suffers from the highest distortion because it mixes the original image with other images. The results in Table II are consistent with our analysis above. The SSIM of VICTOR is the highest, followed by ML-DA, while MT is significantly lower than both.

Table III provides the overall auditing performance on the four evaluation metrics of various methods. We make the following observations. First, VICTOR consistently outperforms the baselines across all evaluation metrics, demonstrating its effectiveness. This is attributed to its ability to amplify the influence of modified samples on the target model, thereby enabling more accurate detection of dataset misuse. Second, the value of Δacc on the HMDB-51 dataset is lower than that on UCF-101. This is likely due to the smaller scale of HMDB-51, where the impact of modified samples is more pronounced. Third, for ML-DA, the TPR is zero in multiple scenarios, and F1 scores cannot even be computed because both precision and recall are zero. On the one hand, ML-DA relies on comparing the predictions of published and unpublished data. This is less effective in the video domain, where models make decisions based on the joint information from multiple frames. On the other hand, ML-DA requires a relatively high proportion of modified samples to be effective. When only a small number of samples are modified, ML-DA struggles to produce meaningful results. Moreover, for MT, the TPR reaches 1 on HMDB-51, but the corresponding FPR remains high. This is because MT determines the decision threshold based on the loss distribution of non-member data on the suspect model. However, the loss of member data on a model that has not actually been trained on the target dataset can still fall below this threshold, leading to frequent misclassification. The issue is further exacerbated when the suspect model is trained from scratch (*i.e.*, SlowFast), as its output tends to be more random. Furthermore, the accuracy of MT drops significantly due to its heavy distortion of the original data. In

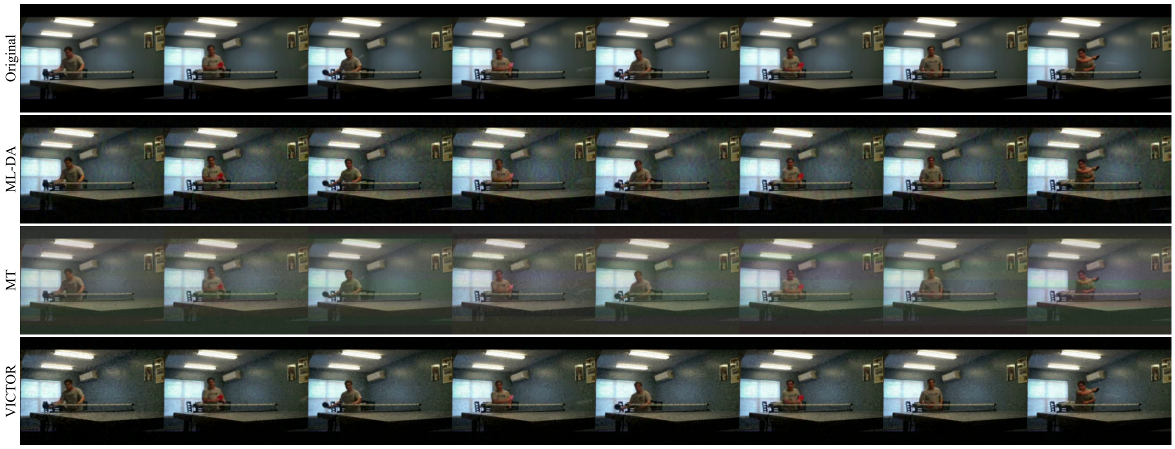

Fig. 4: An illustration of generated videos of different methods.

TABLE III: Overall auditing performance on the evaluation metrics.

Dataset	Model / Method / Metric	I3D			SlowFast			TSM		
		ML-DA [16]	MT [43]	VICTOR	ML-DA [16]	MT [43]	VICTOR	ML-DA [16]	MT [43]	VICTOR
HMDB-51	Δacc	-0.063	-0.065	**-0.020**	-0.072	-0.052	**-0.042**	-0.043	-0.044	**-0.032**
	TPR	0.400	**1.000**	**1.000**	0.000	**1.000**	**1.000**	0.000	**1.000**	**1.000**
	F1 Score	0.308	0.500	**1.000**	N/A	0.294	**1.000**	N/A	0.357	**1.000**
	FPR	0.120	0.200	**0.000**	**0.000**	0.480	**0.000**	0.100	0.360	**0.000**
UCF-101	Δacc	-0.024	-0.038	**-0.021**	-0.014	-0.036	**-0.012**	-0.015	-0.038	**-0.011**
	TPR	0.000	0.700	**1.000**	0.000	0.800	**1.000**	0.000	0.800	**1.000**
	F1 Score	N/A	0.609	**1.000**	N/A	0.229	**1.000**	N/A	0.889	**1.000**
	FPR	0.040	0.060	**0.000**	0.210	0.520	**0.000**	**0.000**	**0.000**	**0.000**

addition, suspect models trained on larger datasets (*i.e.*, UCF-101) generally exhibit stronger generalization, which reduces the loss gap between member and non-member samples and leads to a decline in TPR. In contrast, VICTOR achieve 100% accuracy on multiple datasets and models, which emphasizes the practicality of VICTOR.

C. Ablation Study

In this section, we conduct ablation studies to evaluate the influence of different components. Specifically, we examine the effectiveness of the evaluation model, threshold clipping, and post processing on the overall auditing performance.

Effectiveness of Evaluation Model. Recalling Section V-A and Section V-B, we adopt an evaluation model to verify the effectiveness of modified samples, which can help the selection of samples. In this section, our aim is to explore the necessity of the evaluation model. Figure 5 illustrates the TPR and FPR results with and without the evaluation model. Here, "Without Evaluation Model" means that the modified and reference samples are randomly selected.

We observe that the verification accuracy is slightly lower when the evaluation model is removed, compared to the case where it is applied. Specifically, the TPR decreases when the suspect model is I3D, and the FPR becomes greater than zero when the suspect model is SlowFast. This is because, without guidance from the evaluation model, the selected modified and reference samples may have less consistent effects on the target model, leading to a slight decline in auditing performance. The above results indicate that the evaluation model is helpful to enhance auditing effectiveness. At the same time, we would like to clarify that VICTOR can still achieve competitive auditing performance without the evaluation model, rather than failing completely.

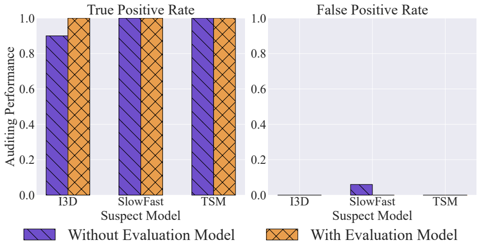

Fig. 5: The effectiveness of evaluation model on the two metrics of the UCF-101 dataset.

Effectiveness of Threshold Clipping. Recalling Section V-C, we introduce a threshold clipping mechanism to mitigate potential false positives. Here, we examine the auditing performance with and without this mechanism. Figure 6 presents the TPR and FPR on the UCF-101 dataset under both settings. The results show that while the TPR remains consistent between the two methods, the FPR is substantially higher when threshold clipping is disabled. This demonstrates that threshold clipping effectively reduces the false positive rate of VICTOR.

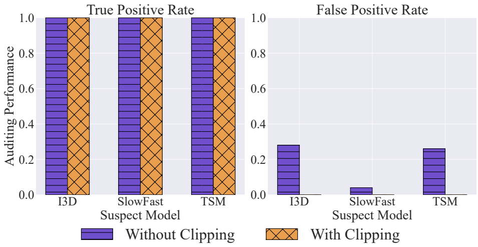

Fig. 6: The effectiveness of threshold clipping on the two metrics of the UCF-101 dataset.

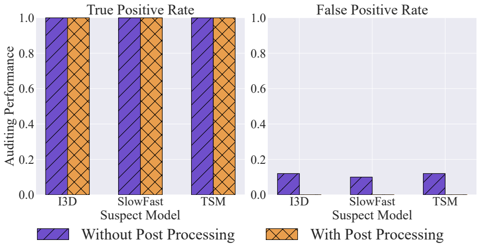

Fig. 7: The effectiveness of post processing on the two metrics of the HMDB-51 dataset.

Moreover, we observe that the impact of threshold clipping is more pronounced for the I3D and TSM models. This is primarily because these models are fine-tuned from pretrained networks, which endows them with stronger generalization capabilities. In this case, their output differences between modified and original samples are smaller than SlowFast.

Effectiveness of Post Processing. During the copyright verification phase, we apply post-processing to the output probability differences. In this section, we investigate the necessity of this post-processing step. Figure 7 presents the auditing results on the HMDB-51 dataset with and without post-processing. Without post-processing, various models exhibit false positives. This is mainly because some suspect models have relatively weak predictive capabilities, leading to low output probabilities for both the unused original samples and their modified counterparts. Consequently, the output differences are also small, which increases the likelihood of misjudgment. In contrast, after applying post-processing, the FPR drops to zero. These results emphasize the key role of post-processing in improving the auditing accuracy of VICTOR.

D. Top-K and Label-only Settings

In this section, we consider two more challenging scenarios (*i.e.*, Top-K and Label-only) to validate the effectiveness of VICTOR. In the Top-K ($K = 5$) setting, only the labels of the top K highest-probability samples are provided. In the Label-only setting, only the label corresponding to the highest-probability sample is output.

Figure 8 and Figure 9 illustrate the auditing performance on the HMDB-51 and UCF-101 datasets. For the Top-K scenario, VICTOR still exhibits great auditing performance across multiple suspect models and datasets (*i.e.*, TPR = 1 and FPR = 0). This highlights the versatility and significant advantages of VICTOR in dataset auditing scenarios. For the Label-only scenario, the TPR of VICTOR in partial cases shows a slight decrease. The reason is that the obtained information under this scenario is limited, posing a huge challenge to accurate auditing. Nevertheless, VICTOR still demonstrates competitive auditing performance and low FPRs.

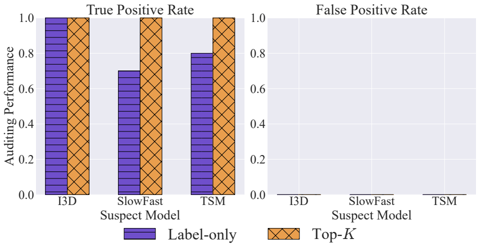

Fig. 8: The auditing performance under Top-K and Label-only settings of the HMDB-51 dataset.

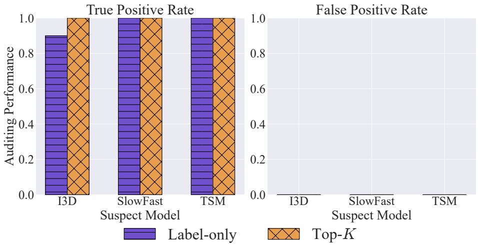

Fig. 9: The auditing performance under Top-K and Label-only settings of the UCF-101 dataset.

E. Parameter Variation

In this section, our aim is to analyze the impact of various parameter settings in VICTOR on the final auditing performance. In particular, we examine the effects of the perturbation budget and the modification ratio in the section. The analysis of the evaluation model and the threshold setting is deferred to Appendix C due to space limitations.

Impact of Perturbation Budget. Figure 10 illustrates the performance of VICTOR on four evaluation metrics on the UCF101 dataset as the perturbation budget ε increases from 2 to 10. We observe that as the budget increases from 2 to 6, the TPR and F1 score consistently improve, the FPR decreases significantly, and Δacc drops slightly. When the budget reaches 6, the auditing accuracy across all three models is already very high. Notably, even with a perturbation budget as low as 2, VICTOR still achieves a TPR exceeding 0.6 and an FPR below 0.2. These results demonstrate that VICTOR possesses strong auditing capabilities and can deliver impressive performance even under a limited perturbation budget.

Impact of Modification Ratio. Figure 11 illustrates the impact of different modification ratios on auditing performance using the UCF101 dataset. It can be observed that when the modification ratio is as low as 0.5%, VICTOR achieves an FPR of 0 across all three suspect models, along with a TPR of 100% on the I3D and SlowFast models. The TPR on the

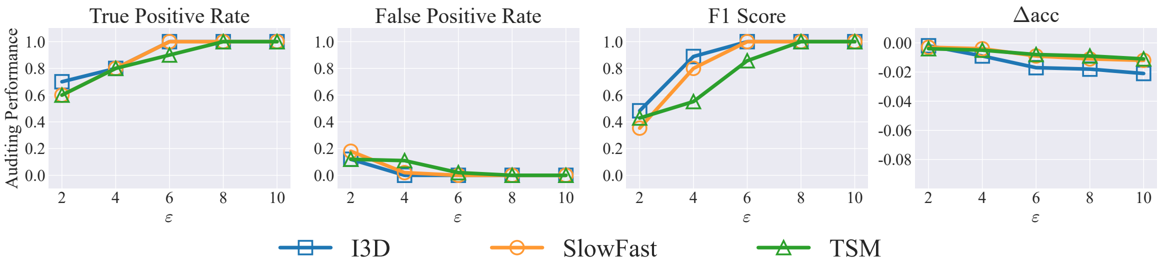

Fig. 10: The impact of perturbation budget for three various suspect models on the four metrics of the UCF-101 dataset.

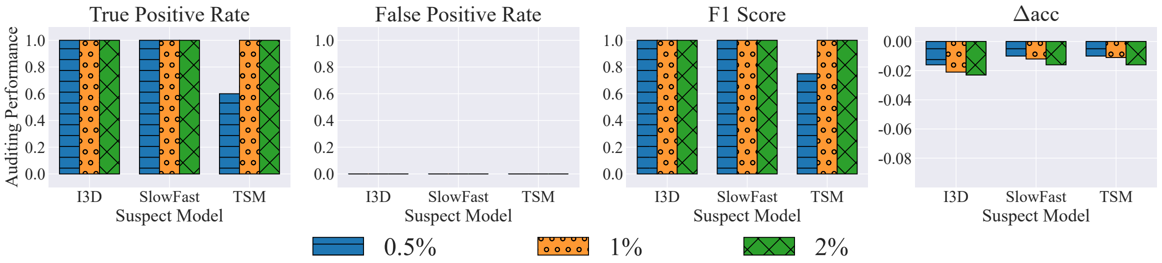

Fig. 11: The impact of modification ratio for three various suspect models on the four metrics of the UCF-101 dataset.

TSM model in this case is 0.6, possibly because TSM is fine-tuned from a pre-trained model with relatively few fine-tuning iterations, making the influence of a low modification ratio less pronounced. Overall, VICTOR can achieve promising auditing performance even at a small modification ratio, showcasing its effectiveness. When the modification ratio increases to 1% or 2%, VICTOR achieves perfect auditing accuracy on all three suspect models (*i.e.*, TPR = 1 and FPR = 0), while maintaining limited impact on normal task performance. These results confirm that VICTOR can achieve high-precision dataset auditing with low modification cost.

F. Robustness

In this section, we investigate the robustness of VICTOR when the pipeline of target models is perturbed to evade auditing. Here, we consider three types of classic mechanisms: input preprocessing, training intervention, post-adjustment.

Input preprocessing. We explore two various approaches to handle the input: input perturbation and input detection. Regarding the input perturbation, this method tries to inject noise into each frame of all samples. Here, the added noise is sampled from a Gaussian distribution with standard deviation $\sigma = 10$. For input detection, this approach tries to identify abnormal samples and remove them from the training. The details of input detection can be found in Appendix C of [71].

TABLE IV: Auditing performance under input perturbation.

Dataset	Model Metric	I3D	SlowFast	TSM
HMDB-51	Δacc	-0.053	-0.068	-0.052
	TPR	1.000	0.900	0.900
	F1 Score	1.000	0.947	0.947
	FPR	0.000	0.000	0.000
UCF-101	Δacc	-0.060	-0.045	-0.028
	TPR	1.000	1.000	1.000
	F1 Score	1.000	1.000	1.000
	FPR	0.000	0.000	0.000

TABLE V: Auditing performance under early stopping.

Dataset	Model Metric	I3D	SlowFast	TSM
HMDB-51	Δacc	-0.162	-0.361	-0.185
	TPR	0.500	0.300	0.400
	F1 Score	0.667	0.462	0.571
	FPR	0.000	0.000	0.000
UCF-101	Δacc	-0.160	-0.388	-0.073
	TPR	1.000	0.600	1.000
	F1 Score	1.000	0.750	1.000
	FPR	0.000	0.000	0.000

Table IV provides the auditing results after introducing the input perturbation. We summarize the following key observations: First, adding input noise has a more pronounced negative impact on the performance of the target model in the normal task. Second, despite the presence of input perturbation, VICTOR still successfully identifies all suspect models on the UCF-101 dataset. On the other hand, the TPR for the SlowFast and TSM models on the HMDB-51 dataset shows a slight decline. This may be attributed to the smaller scale of the HMDB-51 dataset, where input noise can have a stronger influence on the optimization of the target model's parameters. Third, VICTOR consistently achieves an FPR of 0 across all settings. This highlights the robustness and reliability of VICTOR in dataset auditing. In practical scenarios, falsely accusing a model of dataset misuse can lead to severe consequences, including reputational harm, financial costs, and strain of judicial resources, particularly in high-stakes intellectual property disputes.

TABLE VI: Auditing performance under post-adjustment.

Dataset	Type	Fine-tuning			Model pruning			Output noise		
	Model Metric	I3D	SlowFast	TSM	I3D	SlowFast	TSM	I3D	SlowFast	TSM
HMDB-51	Δacc	-0.032	-0.018	-0.047	-0.370	-0.396	-0.424	-0.181	-0.287	-0.222
	TPR	0.800	1.000	0.900	0.500	0.100	0.600	1.000	0.800	0.900
	F1 Score	0.889	1.000	0.947	0.667	0.182	0.750	1.000	0.889	0.947
	FPR	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
UCF-101	Δacc	-0.041	-0.002	-0.011	-0.593	-0.582	-0.616	-0.156	-0.185	-0.126
	TPR	1.000	0.900	1.000	1.000	0.900	1.000	1.000	1.000	1.000
	F1 Score	1.000	0.947	1.000	1.000	0.947	1.000	1.000	1.000	1.000
	FPR	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Training Intervention. In this countermeasure, a malicious attacker can train the video recognition model for a small number of epochs to prevent the model from overfitting the training samples (*i.e.*, early stopping). Here, we choose to train the models only for 10 epochs.

Table V presents the auditing performance on the two datasets when early stopping is applied. We make the following observations: First, the normal performance of all three target models declines significantly under early stopping, with the SlowFast model experiencing the most severe drop. This is primarily because SlowFast is trained from scratch without the benefit of pretrained weights, and thus requires more training epochs to learn and update the model parameters. In this case, VICTOR demonstrates better auditing performance on the I3D and TSM models compared to the SlowFast model. For example, on the UCF-101 dataset, VICTOR achieves a TPR of 1.0 for both I3D and TSM, while the TPR for SlowFast drops to 0.6. In addition, we find that VICTOR shows more effective auditing on the UCF-101 dataset than on the HMDB51 dataset. This is likely due to the larger scale of UCF-101, which allows the target model to learn more knowledge within the same number of training rounds, thereby enabling VICTOR to more accurately detect dataset misuse. Actually, when early stopping is applied, the accuracy for the video models like SlowFast on the HMDB-51 dataset is very low (*i.e.*, around 20%). This indicates that the model has not fully captured the characteristics of the dataset, making the auditing effect less than ideal. At the same time, it should also be noted that the inherent value of this type of model is limited. Finally, VICTOR maintains an FPR of 0 across all datasets and models, demonstrating its reliability and practicality.

Post Adjustment. Table VI provides the auditing performance under various post-adjustment methods. First, we investigate the impact of model fine-tuning as a potential evasion strategy. Specifically, we assume that a malicious attacker possesses a dataset with a distribution similar to the published dataset (approximately 10% of the original dataset's scale) and uses it to fine-tune the suspect model.

The three columns corresponding to "Fine-tuning" report the performance of VICTOR across four evaluation metrics on the fine-tuned models. First, we observe that the performance of the SlowFast model on normal tasks improves after fine-tuning. This is mainly because the SlowFast model is trained from scratch, and exposure to a dataset with a similar distribution can further enhance its learning. Second, the TPR of VICTOR shows a slight decline on some scenarios. This is attributed to the fine-tuning process, which diminishes the influence of the original published dataset on the target model, thereby reducing the detectability of dataset misuse. Finally, VICTOR consistently achieves an FPR of 0 across all settings. Overall, the above results and analysis suggest that VICTOR keeps promising performance against model fine-tuning.

Next, we explore the influence of another evasion strategy (*i.e.*, model pruning). The middle three columns (corresponding to "Model pruning") in Table VI present the auditing performance of VICTOR on the HMDB51 and UCF-101 datasets when the pruning ratio is set to 40%. We observe that the pruned models exhibit a substantial performance drop on the normal task. Under this condition, the TPR of VICTOR decreases significantly on the HMDB-51 dataset, while it remains high on the UCF-101 dataset. This discrepancy is primarily due to the smaller scale of the HMDB-51 dataset, which limits the performance of the trained model. After pruning, the test accuracy on HMDB-51 becomes extremely low, posing considerable challenges to the audit process. In contrast, although model performance on UCF-101 also declines after pruning, it remains superior to that of the pruned models on HMDB-51, enabling VICTOR to maintain high auditing effectiveness. Moreover, VICTOR still achieves zero false positives across all three models on the two datasets. In summary, while model pruning can reduce detection effectiveness to some extent, this operation also severely impairs the model's performance on normal tasks, limiting its practicality.

Finally, we investigated the impact of introducing noise into the model's output. Specifically, we injected zero-mean Gaussian noise with a standard deviation of 0.1 into the model outputs to simulate this disturbance. The last three columns (*i.e.*, "Output noise") in Table VI present the auditing results of VICTOR under this setting. On the one hand, the injected noise significantly degrades the model's performance on the original normal task. On the other hand, VICTOR maintains 100% auditing accuracy on the UCF-101 dataset, while its TPR exhibits a slight decline on the HMDB-51 dataset. This difference can be attributed to the varying scales of these two datasets, which lead to differences in the performance of the corresponding target models. These results demonstrate that

VICTOR remains robust even in the presence of output noise, further validating its effectiveness in practical audit scenarios.

VII. Discussion

VICTOR is the first auditing mechanism designed for video data that verifies whether a deployed video recognition model was trained on the specific dataset. At the same time, VICTOR still has several limitations that warrant further investigation. First, black-box query access may be unavailable for fully offline-deployed systems. Further, sophisticated adversaries aware of our auditing mechanism could develop adaptive evasion strategies. Therefore, evaluating robustness against more sophisticated adaptive evasion strategies and reducing computational overhead for scalable deployment will be valuable future work.

VIII. Conclusion

In this paper, we propose VICTOR, a method for verifying whether a target video dataset is used to train a suspect recognition model. The core idea is to amplify the influence of published modified samples on the prediction behavior of the target model. By injecting carefully designed noise and selecting specific samples, combined with key hypothesis testing, VICTOR can achieve 100% auditing accuracy across two datasets and three target models while keeping the modifications nearly imperceptible. Extensive ablation experiments validate the effectiveness of various components. Furthermore, we also explore the impact of different parameter settings. Finally, we evaluate the robustness of proposed VICTOR,

Ethics Considerations

This paper focuses on dataset copyright auditing in video recognition systems. We strictly followed ethical guidelines by using publicly available, open-source datasets, under licenses that permit research and educational use. As these datasets were curated and released by third parties, direct informed consent was not applicable. However, we are committed to ethical data use and will comply with all licensing terms for any future modifications or redistribution. We aim to advance technological development while upholding academic ethics and data copyright norms, thereby supporting a healthy research community and fostering a culture of lawful and compliant dataset sharing.

Acknowledgments

We thank the anonymous reviewers for their constructive and insightful feedback. This work is supported in part by the National Natural Science Foundation of China under Grants No. (62402431, 62441618, 62025206, U23A20296, U24A20237, 62402379, 72594583011, 7257010373), Key R&D Program of Zhejiang Province under Grants No. (2024C01259, 2025C01061, 2024C01065, 2024C01012, 2025C01089), the project CiCS of the research programme Gravitation which is (partly) financed by the Dutch Research Council (NWO) under Grant 024.006.037, the China Postdoctoral Science Foundation under Grants No. (2025M771501, BX20250380), and Zhejiang University.

References

[1] F. Wu, Q. Wang, J. Bian, N. Ding, F. Lu, J. Cheng, D. Dou, and H. Xiong, "A Survey on Video Action Recognition in Sports: Datasets, Methods and Applications," *IEEE Transactions on Multimedia*, vol. 25, pp. 7943–7966, 2022.

[2] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, "Content-Based Video Recommendation System Based on Stylistic Visual Features," *Journal on Data Semantics*, 2016.

[3] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human Action Recognition from Various Data Modalities: A Review," *IEEE TPAMI*, vol. 45, no. 3, pp. 3200–3225, 2022.

[4] M. Biparva, D. Fernández-Llorca, R. I. Gonzalo, and J. K. Tsotsos, "Video Action Recognition for Lane-Change Classification and Prediction of Surrounding Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 569–578, 2022.

[5] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, "A Review of Video Surveillance Systems," *Journal of Visual Communication and Image Representation*, vol. 77, p. 103116, 2021.

[6] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The Kinetics Human Action Video Dataset," *CoRR abs/1705.06950*, 2017.

[7] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding," in *ECCV*, 2016, pp. 510–526.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li, "Large-Scale Video Classification with Convolutional Neural Networks," in *CVPR*, 2014, pp. 1725–1732.

[9] A. Gilbertson and A. Reisner, "Apple, Nvidia, Anthropic Used Thousands of Swiped YouTube Videos to Train AI," https://www.proofnews.org/apple-nvidia-anthropic-used-thousands-of-swiped-youtube-videos-to-train-ai, 2024.

[10] A. Sablayrolles, M. Douze, C. Schmid, and H. Jégou, "Radioactive Data: Tracing through Training," in *ICML*, 2020, pp. 8326–8335.

[11] Y. Li, M. Zhu, X. Yang, Y. Jiang, T. Wei, and S.-T. Xia, "Black-Box Dataset Ownership Verification via Backdoor Watermarking," *IEEE TIFS*, vol. 18, pp. 2318–2332, 2023.

[12] J. Guo, Y. Li, L. Wang, S.-T. Xia, H. Huang, C. Liu, and B. Li, "Domain Watermark: Effective and Harmless Dataset Copyright Protection is Closed at Hand," in *NeurIPS*, 2023, pp. 54 421–54 450.

[13] H. Guo, J. Guo, B. Chen, Y. Wang, X. Chen, H. Huang, Q. Yan, and L. Xiao, "AUDIO WATERMARK: Dynamic and Harmless Watermark for Black-Box Voice Dataset Copyright Protection," in *USENIX Security*, 2025.

[14] Y. Miao, M. Xue, C. Chen, L. Pan, J. Zhang, B. Z. H. Zhao, D. Kaafar, and Y. Xiang, "The Audio Auditor: User-Level Membership Inference in Internet of Things Voice Services," in *PoPETS*, 2021.

[15] L. Du, X. Zhou, M. Chen, C. Zhang, Z. Su, P. Cheng, J. Chen, and Z. Zhang, "SoK: Dataset Copyright Auditing in Machine Learning Systems," in *IEEE S&P*, 2025.

[16] Z. Huang, N. Z. Gong, and M. K. Reiter, "A General Framework for Data-Use Auditing of ML Models," in *ACM CCS*, 2024.

[17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *IEEE S&P*, 2017, pp. 3–18.

[18] C. Song and V. Shmatikov, "Auditing Data Provenance in Text-Generation Models," in *SIGKDD*, 2019, pp. 196–206.

[19] M. Chen, Z. Zhang, T. Wang, M. Backes, and Y. Zhang, "FACE-AUDITOR: Data Auditing in Facial Recognition Systems," in *USENIX Security*, 2023, pp. 7195–7212.

[20] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li, "Untargeted Backdoor Watermark: Towards Harmless and Stealthy Dataset Copyright Protection," in *NeurIPS*, 2022.

[21] K. T. Co, L. Muñoz-González, S. de Maupeou, and E. C. Lupu, "Procedural Noise Adversarial Examples for Black-Box Attacks on Deep Convolutional Networks," in *ACM CCS*, 2019, pp. 275–289.

[22] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership Inference Attacks from First Principles," in *IEEE S&P*, 2022, pp. 1897–1914.

[23] Z. Zhang, M. Chen, M. Backes, Y. Shen, and Y. Zhang, "Inference Attacks Against Graph Neural Networks," in *USENIX Security Symposium*, 2022, pp. 4543–4560.

[24] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang, "ML-Doctor: Holistic Risk Assessment of

Inference Attacks Against Machine Learning Models," in *USENIX Security Symposium*, 2022, pp. 4525–4542.

[25] P. Maini, M. Yaghini, and N. Papernot, "Dataset Inference: Ownership Resolution in Machine Learning," in *ICLR*, 2021.

[26] Z. Li and Y. Zhang, "Membership Leakage in Label-Only Exposures," in *ACM CCS*, 2021, pp. 880–895.

[27] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-Only Membership Inference Attacks," in *ICML*, 2021.

[28] Z. Tian, Z. Wang, A. M. Abdelmoniem, G. Liu, and C. Wang, "Knowledge Representation of Training Data with Adversarial Examples Supporting Decision Boundary," *IEEE TIFS*, 2023.

[29] S. Szyller, R. Zhang, J. Liu, and N. Asokan, "On the Robustness of Dataset Inference," *TMLR*, 2023.

[30] L. Du, M. Chen, M. Sun, S. Ji, P. Cheng, J. Chen, and Z. Zhang, "ORL-AUDITOR: Dataset Auditing in Offline Deep Reinforcement Learning," in *NDSS*, 2024.

[31] H. Liu, J. Jia, W. Qu, and N. Z. Gong, "EncoderMI: Membership Inference Against Pre-trained Encoders in Contrastive Learning," in *ACM CCS*, 2021, pp. 2081–2095.

[32] L. Song and P. Mittal, "Systematic Evaluation of Privacy Risks of Machine Learning Models," in *USENIX Security Symposium*, 2021, pp. 2615–2632.

[33] L. Du, Z. Zhu, M. Chen, Z. Su, S. Ji, P. Cheng, J. Chen, and Z. Zhang, "ArtistAuditor: Auditing Artist Style Pirate in Text-to-Image Generation Models," in *WWW*, 2025, pp. 2500–2513.

[34] Q. Li, R. Yu, and X. Wang, "Vid-SME: Membership Inference Attacks Against Large Video Understanding Models," *CoRR abs/2506.03179*, 2025.

[35] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou, "White-Box vs Black-Box: Bayes Optimal Strategies for Membership Inference," in *ICML*, 2019, pp. 5558–5567.

[36] A. Dziedzic, H. Duan, M. A. Kaleem, N. Dhawan, J. Guan, Y. Cattan, F. Boenisch, and N. Papernot, "Dataset Inference for Self-Supervised Models," in *NeurIPS*, 2022.

[37] G. Li, S. Rezaei, and X. Liu, "User-Level Membership Inference Attack Against Metric Embedding Learning," in *ICLR PAIR2Struct Workshop*, 2022.

[38] T. Dong, S. Li, G. Chen, M. Xue, H. Zhu, and Z. Liu, "RAI2: Responsible Identity Audit Governing the Artificial Intelligence," in *NDSS*, 2023.

[39] G. Liu, T. Xu, X. Ma, and C. Wang, "Your Model Trains on My Data? Protecting Intellectual Property of Training Data via Membership Fingerprint Authentication," *IEEE TIFS*, vol. 17, pp. 1024–1037, 2022.

[40] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models," in *NDSS*, 2019.

[41] E. Wenger, X. Li, B. Y. Zhao, and V. Shmatikov, "Data Isotopes for Data Provenance in DNNs," in *PoPETS*, 2024, pp. 413–429.

[42] J. Guo, Y. Li, R. Chen, Y. Wu, C. Liu, and H. Huang, "ZeroMark: Towards Dataset Ownership Verification without Disclosing Watermark," in *NeurIPS*, 2024.

[43] Z. Chen and K. Pattabiraman, "Anonymity Unveiled: A Practical Framework for Auditing Data Use in Deep Learning Models," in *ACM CCS*, 2025.

[44] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[45] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible Backdoor Attack with Sample-Specific Triggers," in *CVPR*, 2021, pp. 16 463–16 472.

[46] Y. Li, Z. Zhang, J. Bai, B. Wu, Y. Jiang, and S.-T. Xia, "Open-Sourced Dataset Protection via Backdoor Watermarking," *CoRR abs/2010.05821*, 2020.

[47] Y. Li, M. Zhu, X. Yang, Y. Jiang, and S.-T. Xia, "Black-Box Ownership Verification for Dataset Protection via Backdoor Watermarking," *CoRR abs/2209.06015*, 2022.

[48] H. A. Al Kader Hammoud, S. Liu, M. Alkhrashi, F. Albalawi, and B. Ghanem, "Look Listen and Attack: Backdoor Attacks Against Video Action Recognition," in *CVPR Workshop*, 2024, pp. 3439–3450.

[49] H. Souri, L. Fowl, R. Chellappa, M. Goldblum, and T. Goldstein, "Sleeper Agent: Scalable Hidden Trigger Backdoors for Neural Networks Trained from Scratch," in *NeurIPS*, 2022.

[50] R. Tang, Q. Feng, N. Liu, F. Yang, and X. Hu, "Did You Train on My Dataset? Towards Public Dataset Protection with Cleanlabel Backdoor Watermarking," in *ACM SIGKDD*, 2023.

[51] W. Bouaziz, N. Usunier, and E.-M. El-Mhamdi, "Data Taggants: Dataset Ownership Verification via Harmless Targeted Data Poisoning," in *ICLR*, 2025.

[52] A. Sadhu, T. Gupta, M. Yatskar, R. Nevatia, and A. Kembhavi, "Visual Semantic Role Labeling for Video Understanding," in *CVPR*, 2021, pp. 5589–5600.

[53] P. Pareek and A. Thakkar, "A Survey on Video-Based Human Action Recognition: Recent Updates, Datasets, Challenges, and Applications," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2259–2322, 2021.

[54] A. Parashar, A. Parashar, W. Ding, M. Shabaz, and I. Rida, "Data Preprocessing and Feature Selection Techniques in Gait Recognition: A Comparative Study of Machine Learning and Deep Learning Approaches," *Pattern Recognition Letters*, vol. 172, pp. 65–73, 2023.

[55] H. Liang, Z. Zhang, C. Hu, Y. Gong, and D. Cheng, "A Survey on Spatio-Temporal Big Data Analytics Ecosystem: Resource Management, Processing Platform, and Applications," *IEEE Transactions on Big Data*, vol. 10, no. 2, pp. 174–193, 2023.

[56] J. Lin, C. Gan, and S. Han, "TSM: Temporal Shift Module for Efficient Video Understanding," in *CVPR*, 2019, pp. 7083–7093.

[57] C. Luo and A. L. Yuille, "Grouped Spatial-Temporal Aggregation for Efficient Action Recognition," in *CVPR*, 2019.

[58] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition," in *ECCV*, 2016, pp. 20–36.

[59] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-Temporal LSTM with Trust Gates for 3d Human Action Recognition," in *ECCV*, 2016.

[60] C. Feichtenhofer, "X3D: Expanding Architectures for Efficient Video Recognition," in *CVPR*, 2020, pp. 203–213.

[61] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," in *CVPR*, 2019, pp. 6202–6211.

[62] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," in *CVPR*, 2018, pp. 6450–6459.

[63] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video Classification with Channel-Separated Convolutional Networks," in *CVPR*, 2019, pp. 5552–5561.

[64] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and The Kinetics Dataset," in *CVPR*, 2017.

[65] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video Swin Transformer," in *CVPR*, 2022, pp. 3202–3211.

[66] G. Bertasius, H. Wang, and L. Torresani, "Is Space-Time Attention All You Need for Video Understanding?" in *ICML*, 2021.

[67] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A Video Vision Transformer," in *CVPR*, 2021, pp. 6836–6846.

[68] A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker, "A Survey of Procedural Noise Functions," in *Computer Graphics Forum*, vol. 29, no. 8, 2010, pp. 2579–2600.

[69] K. Perlin, "Improving Noise," in *ACM SIGGRAPH*, 2002.

[70] R. F. Woolson, "Wilcoxon Signed-Rank Test," *Encyclopedia of Biostatistics*, vol. 8, 2005.

[71] Q. Yuan, Z. Zhang, L. Du, M. Chen, M. Sun, Y. Gao, S. He, and J. Chen, "VICTOR: Dataset Copyright Auditing in Video Recognition Systems," *CoRR abs/2512.14439*, 2025.

[72] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition," in *ICCV*, 2011, pp. 2556–2563.

[73] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild," *CoRR/abs:1212.0402*, 2012.

[74] R. Goyal *et al.*, "The 'Something Something' Video Database for Learning and Evaluating Visual Common Sense," in *ICCV*, 2017, pp. 5842–5850.

[75] C.-F. R. Chen, R. Panda, K. Ramakrishnan, R. Feris, J. Cohn, A. Oliva, and Q. Fan, "Deep Analysis of Cnn-Based Spatio-Temporal Representations for Action Recognition," in *CVPR*, 2021.

[76] M. Contributors, "OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark," https://github.com/open-mmlab/mmaction2, 2020.

[77] K. Jordan, "On the Variance of Neural Network Training with respect to Test Sets and Distributions," in *ICLR*, 2024.

Appendix

A. Threshold Analysis

Theorem 1 (Threshold range under TPR and FPR Constraints)**.** *Let $\bar{h}$ denote the average probability difference between two models over n samples. Assume that under the null hypothesis (i.e., same training dataset), $\bar{h} \sim \mathcal{N}(\mu_0, \sigma_0^2/n)$; and under the alternative hypothesis (i.e., different training datasets), $\bar{h} \sim \mathcal{N}(\mu_1, \sigma_1^2/n)$, with $\mu_0 < \mu_1$. For given significance levels a (FPR) and b (FNR), any threshold τ satisfying*

$$\tau \in [\mu_0 + z_{1-b}\frac{\sigma_0}{\sqrt{n}}, \mu_1 + z_a\frac{\sigma_1}{\sqrt{n}}]$$

ensuring that $TPR = P(\bar{h} \le \tau | H_0) \ge 1 - b$ and $FPR = P(\bar{h} \le \tau | H_1) \le a$.

Proof. The TPR and FPR can be expressed as follows:

$$TPR(\tau) = P(\bar{h} \le \tau | H_0) = \Phi(\frac{\tau - \mu_0}{\sigma_0/\sqrt{n}}),$$
$$FPR(\tau) = P(\bar{h} \le \tau | H_1) = \Phi(\frac{\tau - \mu_1}{\sigma_1/\sqrt{n}}),$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. According to the constraints of $TPR \ge 1 - b$ and $FPR \le a$, we can obtain

$$\frac{\tau - \mu_0}{\sigma_0/\sqrt{n}} \ge z_{1-b},$$
$$\frac{\tau - \mu_1}{\sigma_1/\sqrt{n}} \le z_a,$$

which directly yield the bounds as follows:

$$\tau_{min} = \mu_0 + z_{1-b}\frac{\sigma_0}{\sqrt{n}}, \tau_{max} = \mu_1 + z_a\frac{\sigma_1}{\sqrt{n}}.$$

□

Determination of the upper threshold. In practice, due to the diversity of video models and datasets, it is difficult to obtain accurate values of the above parameters. According to existing research [61, 64, 77], the accuracy of training with the same dataset fluctuates by about 1%-2%, while the migration between different training datasets can cause the accuracy deviation to exceed 5% or even 10%. Therefore, here we assume that $\mu_0 = 0.02, \sigma_0 = 0.01, \mu_1 = 0.08, \sigma_1 = 0.02, n = 100$, let $a = b = 0.05$, we can calculate $\tau_{min} \approx 0.022$ and $\tau_{max} \approx 0.077$. To achieve a suitable trade-off between TPR and FPR, we choose the upper threshold limit H as the midpoint between the minimum and maximum values, *i.e.*, $H = (\tau_{min} + \tau_{max})/2 \approx 0.05$.

B. Discussion of Other Methods

In this work, we apply the SOTA auditing methods (*i.e.*, ML-DA [16] and MT [43]) for image data to the video domain as the baselines. For these methods, we process each frame in the video separately and then merge them together.

The idea of ML-DA is to generate two images as far apart as possible for the same sample and then randomly select one to publish, and finally use the suspect model's output on the two generated images for auditing. On the one hand, this method requires a large modification ratio in the published version. On the other hand, video models are more robust in terms of recognition accuracy, making this method less effective.

For MT, a data marking technique based on image blending and noise injection is designed to mark target data. Then, a membership inference-based inference process is utilized to achieve the verification by observing the loss values. Though MT only requires to mark a small fraction of data, it obviously degrades the marked image quality, which reduces the stealthiness. In addition, due to the complexity of the video model and the diversity of video data, the outputs of member samples and some non-member samples are prone to present similar outputs (such as both high or low), which can interfere with the audit results. Moreover, for MT, calculating the threshold based on the non-member loss distribution cannot consistently yield promising results across different video models and datasets. Furthermore, video data consists of a sequence of frames, and injecting different noise into consecutive frames independently can easily disrupt the coherence of the video.

Many gradient optimization-based auditing methods [10, 20] designed for image data are difficult to directly apply to video. This is because raw videos typically undergo frame sampling and cropping, which disrupt the input continuity. As a result, generating effective perturbations for the entire video based on limited input information becomes highly challenging. In addition, many backdoor-based methods require modifying the true labels of the original data, which is often impractical in real-world scenarios. Such approaches are prone to introducing harmful side effects and potential security risks.

C. Parameter Variation

1) Impact of Perturbation Budget: Figure 12 illustrates the impact of various perturbation budgets for three suspect models on the HMDB-51 dataset. We find that even with a perturbation budget of only 2, VICTOR could still achieve a TPR of 1 on the I3D and SlowFast models while maintaining a low FPR. This is because the HMDB51 dataset is relatively small in scale, and the target model's performance is generally lower than on the UCF-101 dataset. In such a setting, VICTOR is more likely to succeed in amplifying the effect of the modified samples. The TSM model exhibits a lower TPR compared to the other two models. We speculate that this is due to fewer training epochs, making the impact of a low perturbation budget less noticeable. As the value of ε increases, the auditing performance of VICTOR continues to improve. When ε reaches 4, VICTOR achieves 100% auditing accuracy on both the I3D and SlowFast models, further demonstrating its effectiveness. Additionally, the modified samples lead to the most significant performance drop on the SlowFast model, likely because it is trained from scratch without a pre-trained backbone, resulting in higher instability during training.

2) Impact of Modification Ratio: Figure 13 presents the auditing performance under different modification ratios on the HMDB-51 dataset. We observe that when the modification ratio is 0.5%, the TPR of the three models does not reach 1,

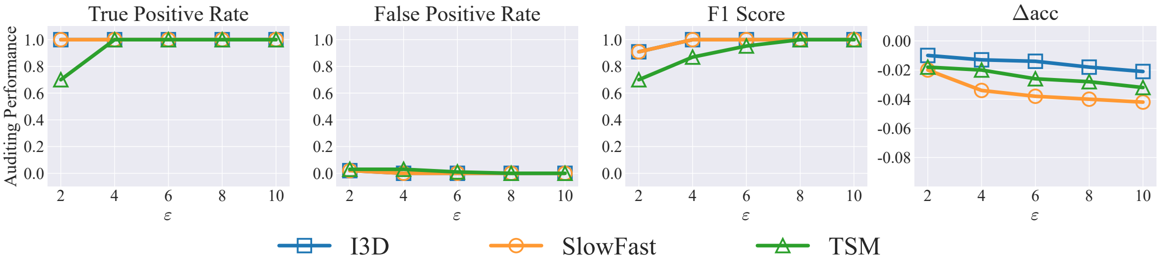

Fig. 12: The impact of perturbation budget for three various suspect models on the four metrics of the HMDB-51 dataset.

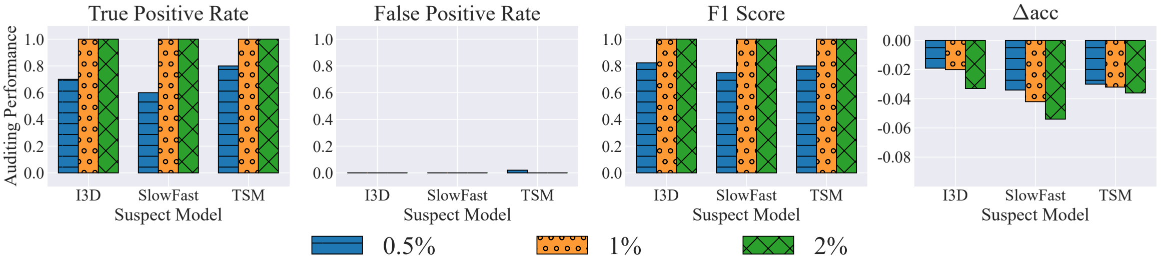

Fig. 13: The impact of modification ratio for three various suspect models on the four metrics of the HMDB-51 dataset.

remaining around 0.6–0.8. This is primarily because HMDB-51 is relatively small in size, and a 0.5% modification ratio corresponds to only about ten samples. Under such conditions, achieving high-precision auditing is challenging, as numerous factors influence model training, and the amplification effect of only a few modified samples tends to be unstable. Nevertheless, VICTOR still achieves a TPR above 0.5 and an FPR below 0.05 across all three suspect models, demonstrating its strong auditing capability. As the modification ratio increases to 1% and 2%, VICTOR can achieve a 100% detection accuracy (*i.e.*, TPR = 1 and FPR = 0), further highlighting its effectiveness.

3) Impact of Various Evaluation Models: Figure 14 presents the results for four metrics on the UCF101 dataset using different evaluation models. We observe that when I3D and TSM are used as evaluation models, both the TPR and FPR achieve optimal performance. In contrast, when SlowFast is applied, the auditing accuracy slightly declines. This is likely because SlowFast is trained from scratch without pre-trained weights, resulting in weaker generalization ability. Consequently, the selected samples are less effective compared to those identified by the other two models. These findings suggest that stronger evaluation models lead to more appropriate sample selection, thereby enhancing the final auditing.

Figure 15 presents the auditing results of different evaluation models for various suspect models on the HMDB-51 dataset. First, we observe that all evaluation models achieve 100% detection accuracy across different suspect models, fully demonstrating the superiority and robustness of VICTOR. Second, we find that when the suspect model is SlowFast, the performance on normal tasks degrades most significantly. This is mainly because, unlike the other two models, SlowFast is trained from scratch without leveraging a pre-trained model, making its training more unstable.

4) Impact of Threshold Setting: As discussed in Section V-C, we apply a threshold clipping mechanism during hypothesis testing. In this section, we examine how different settings of the threshold upper bound H affect the audit results. Since VICTOR performs well under various thresholds when the perturbation budget ε is 10, here we focus on a more challenging scenario and present the changes in TPR and FPR for three suspect models on the UCF-101 dataset when the perturbation budget ε is set to 4.

As shown in Figure 16, when H is set to a very small value (*e.g.*, 0), it imposes a strict condition where the output probability of the original sample must be significantly lower than that of the modified sample for a dataset misuse to be detected. In this case, the FPR of all three suspect models is 0, but the TPR drops to 0.6. Conversely, when H is set to a larger value (*e.g.*, 0.2), the TPR increases, but the FPR also rises significantly. These results indicate that setting H too low or too high leads to suboptimal auditing outcomes, *i.e.*, either a reduced detection rate (low TPR) or increased false alarms (high FPR). Therefore, selecting an appropriate value for H is critical to balancing TPR and FPR. We find that $H = 0.05$ provides a good trade-off between these two aspects, which is consistent with our analysis in Appendix A. Therefore, we adopt this value as the threshold upper bound.

Figure 17 presents the TPR and FPR results under different threshold upper bounds on the HMDB-51 dataset. Consistent with the trends observed on the UCF-101 dataset, we found that when the threshold upper bound H is set to a very small value (*e.g.*, 0), the FPR remains very low, but the TPR drops significantly. As the threshold increases, the TPR improves notably, but at the cost of a higher FPR. To strike a good balance between TPR and FPR across different datasets and suspect models, we consistently set $H = 0.05$ in the experiments.

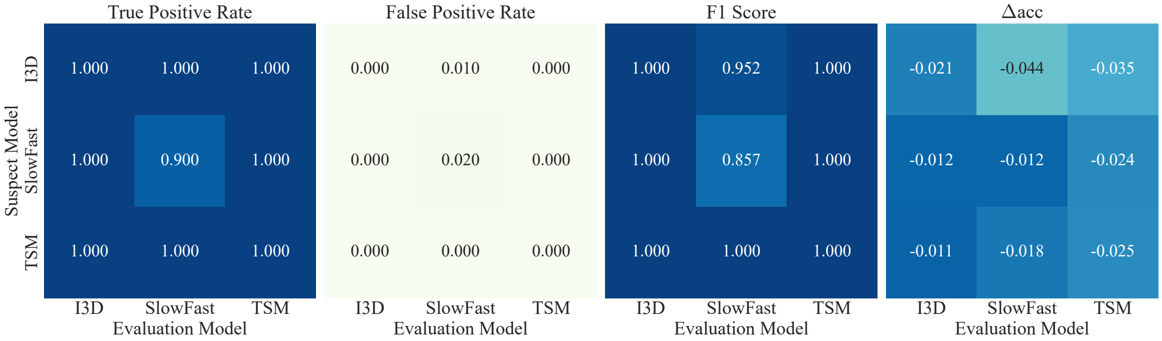

Fig. 14: The impact of different evaluation models for three suspect models on four metrics of the UCF-101 dataset.

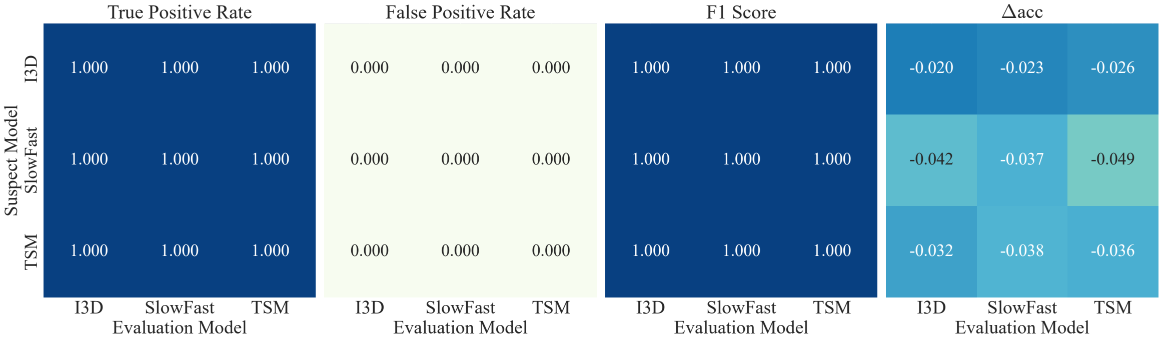

Fig. 15: The impact of different evaluation models for three suspect models on four metrics of the HMDB-51 dataset.

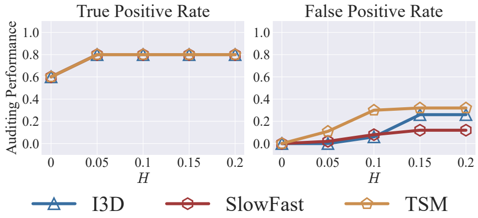

Fig. 16: The impact of different threshold uppers for three various suspect models on the two metrics of the UCF-101 dataset when the noise perturbation budget $\varepsilon = 4$.

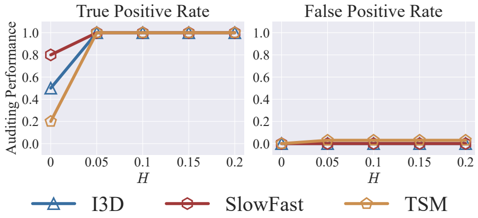

Fig. 17: The impact of different threshold uppers for three various suspect models on the two metrics of the HMDB-51 dataset when the noise perturbation budget $\varepsilon = 4$.

D. Results on Larger Dataset and Backbones

In this section, we explore the effectiveness of VICTOR on a larger dataset (*i.e.*, SSv2) and the transformer-based backbone (*i.e.*, TimeSformer).

TABLE VII: Auditing performance for the TimeSformer model on the SSv2 dataset.

Metric	TPR	FPR	F1	Δacc
Result	1.000	0.000	1.000	-0.003

TABLE VIII: Auditing performance for the TimeSformer model on the SSv2 dataset under Top-K and Label-only settings.

Setting	TPR	FPR
Top-K	1.000	0.000
Label-only	1.000	0.001

Table VII provides the overall auditing performance of VICTOR. It can be seen that VICTOR still achieves perfect auditing results on the large datasets, which highlights the versatility of VICTOR. Furthermore, we also explore the auditing performance under Top-K ($K = 5$) and Label-only settings. As shown in Table VIII, VICTOR still exhibits great auditing accuracy. VICTOR achieves 100% auditing accuracy with the Top-K setting, while a slight decrease in auditing performance occurs with the Label-only setting. This indicates that VICTOR remains robust to stricter constraints.