Unshaken by Weak Embedding: Robust Probabilistic Watermarking for Dataset Copyright Protection

Shang Wang[*], Tianqing Zhu[†✉], Dayong Ye[†], Hua Ma[‡], Bo Liu[*], Ming Ding[‡], Shengfang Zhai[§], Yansong Gao[¶✉]
[*]University of Technology Sydney, Australia. Email: shang.wang-1@student.uts.edu.au; bo.liu@uts.edu.au
[†]City University of Macau, Macau SAR, China. Email: tqzhu@cityu.edu.mo; dayongye@outlook.com
[‡]Data61, CSIRO, Australia. Email: mary.ma@data61.csiro.au; ming.ding@data61.csiro.au
[§]National University of Singapore, Singapore. Email: shengfang.zhai@nus.edu.sg
[¶]School of Cyber Science and Engineering, Southeast University, China. Email: gao.yansong@hotmail.com
T. Zhu and Y. Gao are the corresponding authors.

***Abstract*—In modern Data-as-a-Service (DaaS) ecosystems, data curators such as data brokerage companies aggregate high-quality data from many contributors and monetize it for deep learning model providers. However, malicious curators can sell valuable data but not inform their original contributors, which violates individual benefits and the law. Intrusive watermarking is one of the state-of-the-art (SOTA) techniques for protecting data copyright, and it detects whether a suspicious model carries the predefined pattern. However, these approaches face numerous limitations: struggle to work under low watermark injection rates ($\leq 1.0\%$); performance degradation; false positives; not robust against watermarking cleansing.**

This work proposes an innovative intrusive watermarking approach, dubbed DIP (Data Intelligence Probabilistic Watermarking), to support dataset ownership verification while addressing the limitations above. It applies a distribution-aware sample selection algorithm, embeds probabilistic associations between watermarked samples and multiple outputs, and adopts a two-fold verification framework that leverages both inference results and their distribution as watermark signals. Extensive experiments on 4 image and 5 text datasets demonstrate that DIP maintains the model's performance, and achieves an average watermark success rate of 89.4% at a 1% injection budget. We further validate that DIP is orthogonal to various watermarked data designs and can seamlessly integrate their strengths. Moreover, DIP proves effective across diverse modalities (image and text) and tasks (regression), with strong performance on generation tasks in large language models. DIP exhibits robustness against various adversarial environments, including 3 based on data augmentation, 3 on data cleansing, 4 on robust training and 3 on collusion-based watermark removal, while existing SOTAs fail. The source code is released at https://github.com/SixLab6/DIP.

I. Introduction

Data are an essential component of artificial intelligence (AI) systems. In 2022, data-centric AI [1] was recognized by Gartner [2] as one of the key emerging trends in data analytics and AI, where data is a critical determinant of model performance, particularly in the training of large language models (LLMs) [3], [4]. However, acquiring high-quality data is non-trivial, requiring significant effort to collect and annotate it. Given that certain dataset acquisition involves domain expertise and data regulations, it is practical for model providers to purchase needed data from professional data curator, such as brokerage companies like Appen [5] and Scale AI [6], rather than individual contributors. For example, clickworkers as data contributors can simply download the Clickworker app [7], make a contribution and earn money from it. In this business Data as a Service (DaaS) scenario, data contributors are informed by the data curator about data usage and are compensated per order requested by model providers, as illustrated in Figure 1.

Unfortunately, as the central entity in the DaaS scenario, the data curator may exploit legitimate business processes to maximize its financial gains. Specifically, while continuing to charge the model provider for data usage, the data curator may withhold payments from data contributors and does not inform them of such transactions. Such misconduct not only compromises the interests of data contributors but also amplifies the risks of data misuse. Therefore, contributors must safeguard their copyrights to prevent the curator's unauthorized use.

State-of-The-Art. Unlike *model copyright* protection [8]–[12], which has been extensively studied, *dataset copyright* protection relies on black-box access without training control, and only a few works have explored dataset ownership verification (DOV). These methods seek to determine whether a suspicious model was trained on a given dataset, using either intrusive or non-intrusive approaches [13]. **For non-intrusive DOV,** methods typically extract unique characteristics from contributed datasets as fingerprints. Examples include DeepTaster [14] and dataset-level membership inference [15]–[17]. However, they require access to model architectures or meticulously crafted auxiliary datasets, which remain key limitations in DaaS scenarios. **As for intrusive DOV**, watermarking methods are leveraged. They embed identifiable signals into

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.2026.241356
www.ndss-symposium.org

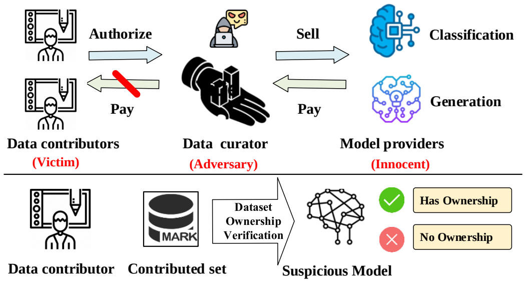


Figure 1: The top panel illustrates the Data as a Service (DaaS), where the data curator outsources data generation and annotation to data contributors and monetizes the data for different model providers. The bottom panel depicts dataset ownership verification against unauthorized use.

Table I: Summary of representative intrusive watermarking works. A fuller circle is more desirable. At the line modality, I = image, T = text; at the line task, C = classification, R = regression, G = generation.

	Style Transformation [18]	Radioactive Data		Backdoor-enabled				
		DW [19]	Data Taggants [20]	DVBW [22]	CBW [23]	UBW [21]	Function--Marker [24]	DIP
Low Watermarking Injection Rate (*RM1*)	○	○	○	◐	◐	○	◐	●
Resilience to Adversarial Environment (*RM2*)	○	◐	◐	○	○	●	○	●
Non-harmful Utility (*RM3*)	◐	◐	●	●	●	◐	●	●
No False Positive (*RM4*)	◐	●	●	●	●	○	●	●
Modality	I	I	I	I, T	I, T	I	T	I, T
Task	C	C	C	C	C	C	G	C, R, G

the contributed dataset through data modification techniques such as style transformations [18], radioactive data [19], [20], and backdoors [21]–[23]. A model trained on the watermarked dataset will exhibit predetermined behaviors on watermarked inputs. Their key advantages include black-box accessibility and resilience to fine-tuning [13], which make them well-suited for DaaS scenarios.

Limitations of Intrusive Watermarking. Despite their promise [13], existing methods exhibit notable limitations in practical DaaS scenarios. **First**, all existing approaches struggle to embed strong watermarks under low watermark injection rates, making watermarked samples unlikely to activate the predefined predictions and thus unreliable for DOV. **Second**, their robustness remains insufficiently examined, raising concerns about reliability when malicious data curators employ adversarial countermeasures. **Third**, style transformation and radioactive data approaches degraded model performance [13], and the untargeted watermarking [21] faces high false positives. **Finally**, all approaches are *restricted to classification tasks*, and most focus exclusively on the image modality.

Requirements & Desirable Properties. Practical and robust watermarking must address these limitations and satisfy the following requirements (RMs). • *RM1: Low Watermark Injection Rate.* Minimal injection rate ($\leq$ 1%) to avoid suspicion [25] while still supporting DOV. • *RM2: Robust to Adversarial Environment.* Resistant to overlooked countermeasures, such as augmentation, cleansing and robust training. • *RM3: Non-harmful Utility.* No degradation of primary task performance. • *RM4: Low False Positive.* Watermarks should only be extracted from watermarked models. • *Desirable Property: Task-Agnostic.* While not a mandatory RM, it is desirable, as existing works focus on classification, leaving other tasks (e.g., regression, generation) unaddressed.

Challenge and Solution. Table I indicates that no watermarking studies can satisfy all the above requirements & property. This motivates our **d**ata **i**ntelligence **p**robabilistic watermarking (DIP) design, to address them. Specifically, we categorize these requirements into three core challenges that DIP must overcome.

•*Challenge 1: Addressing RM1&3.* Low injection budgets exacerbate weak watermark signals, particularly in approaches with complex mappings. DIP addresses this limitation via an effective probabilistic mapping that reduces dependence on injection budget while preserving the original data distribution, thus maintaining model utility. Furthermore, following Shao *et al.* [26], intrusive watermarking often relies on 0-bit signals—the presence or absence of predefined behaviors—for DOV. DIP enhances robustness through a two-fold verification that leverages prediction distributions on watermarked samples as auxiliary evidence, reducing the need for high injection rates.

•*Challenge 2: Addressing RM2.* Adversarial environments may weaken or remove watermarks, preventing the expected behaviors on watermarked samples. DIP mitigates this by breaking the deterministic feature mapping assumed in existing countermeasures: its probabilistic pattern associates watermarked samples with multiple features, preserving robustness where most backdoor-based approaches fail (Section VI-B).

•*Challenge 3: Addressing RM4.* DIP ensures high specificity by using multiple target outputs and a predefined distribution as verification signals, enabling reliable detection exclusively on watermarked inputs and thus satisfying *RM4*.

Overall, DIP embeds a probabilistic watermark by slightly modifying the training data. Models trained on it produce specific prediction distributions (e.g., 30% class A, 70% class B) over watermarked samples. Then, DIP performs DOV through a two-fold verification method, using both predictions and their distribution as signals.

Contribution. Our main contributions are as follows.

• We propose DIP, a probabilistic watermarking framework for DOV that satisfies the above requirements and property. It integrates distribution-aware sample selection, probabilistic watermark injection, and two-fold verification, achieving robust and reliable ownership verification.

• Under low injection rates, we evaluate DIP on 4 image and 5 text datasets. Experiments show DIP preserves model utility, achieves an 89.4% watermark success rate. It remains robust against 13 adversarial settings, including 3 based on data augmentation, 3 on data cleansing, 4 on robust training and 3 on collusion-based watermark removal. Across these settings, DIP significantly outperforms SOTA intrusive approaches.

• We confirm the generalization of DIP. First, it is orthogonal to various watermarked data designs, such as dynamic trigger and OOD data. Second, DIP extends to non-classification tasks, including generation in the context of LLMs and regression. To our knowledge, prior DOV approaches rarely generalize beyond the classification setting, whereas DIP does.

II. Related Work

A. Intrusive Data Copyright Protection

As the main means of DOV, intrusive watermarking approaches apply data poisoning techniques. They generate watermarked samples using specific transformations (e.g., style, radioactive data, backdoors), and associate these samples with the predefined outputs. Models trained on such data produce expected predictions over watermarked inputs, enabling ownership verification through hypothesis testing.

For style transformations, Zou *et al.* [18] converted the original images from RGB to YIQ space and embedded watermarks through hue rotation, preserving semantic content. DOV compares losses on original and watermarked samples, with minimized loss indicating training on watermarked data.

For radioactive data approaches, Guo *et al.* [19] employed a hardly-generalized domain for the original dataset to generate watermarked data, that is DW. Models trained on the dataset containing domain-specific samples can correctly classify the modified inputs specified by the data owner. Bouaziz *et al.* [20] proposed Data Taggants, which uses randomly generated OOD samples paired with random labels as signature keys. By applying clean-label data poisoning [23], these keys are embedded into a small subset of the dataset, enabling any model trained on the signed data to predict the designated labels when queried with the OOD keys.

For backdoor-enabled watermarking, Li *et al.* introduced DVBW, which applies a simple trigger (a small patch) to induce a single-target backdoor. Injecting a few such samples embeds the watermark, forcing trained models to classify triggered inputs as the target label. Watermark injection can be performed under either dirty-label or clean-label settings. Tang *et al.* [23] proposed a similar clean-label approach, denoted CBW. Li *et al.* further presented an untargeted backdoor watermark [21] (UBW), which averages the model's prediction probability of each class conditioned on the watermarked images. For text generation tasks, FunctionMarker [24] constructs a set of function-specific knowledge, comprising custom function names and their corresponding expressions. An LLM trained on this watermarked corpus will produce the predefined function expressions when queried with these function names, thereby enabling watermark extraction.

B. Non-intrusive Data Copyright Protection

Unlike intrusive approaches, non-intrusive dataset protection avoids modifying the original data and instead identifies inherent fingerprints or leverages unique characteristics—often with the help of an auxiliary dataset—for DOV.

DeepTaster [14] trains a one-class meta-classifier using adversarial spectrum images derived from multiple architectures trained on the protected dataset. In addition, several studies [15], [16] explore dataset-level membership inference. For instance, Maini *et al.* [15] estimate the distance of multiple data points to the decision boundary, implementing copyright claims. Dong *et al.* [27] observed that similar model outputs on certain inputs indicate shared training data. To quantify this, they simulate varying overlap ratios by constructing controlled subsets of the protected dataset and training shadow models. A lookup table is then built to map output differences to dataset overlap levels.

C. Limitations for Practical DaaS Scenarios

In practical DaaS scenarios, dataset protection approaches must satisfy *RM1-RM4*, which are essential for ensuring the effectiveness and robustness of watermarking. For non-intrusive protection, DeepTaster requires access to model architectures to train a meta-classifier, which is often not held. With non-overlapping datasets and independently trained models, dataset-level inference approaches often falsely signal dataset theft [28]. As reported in [13], several non-intrusive approaches also rely on intermediate representations, limiting their applicability under black-box verification. For intrusive protection, the style transformation-based approach and DW leverage complex OOD features as watermarks. However, these features demand a high injection rate to be learned, which may alter the original dataset distribution and degrade performance on the main task (*violating RM1 &3*). In black-box settings, the radioactive data approach relies on model extraction to obtain a distilled version of the suspicious model. However, this process is query-intensive and impractical at scale. In backdoor-enabled watermarking, single-target backdoors are easily removed by various adversarial environments, such as ABL [29], and ASSET [30] (*violating RM2*). Although untargeted backdoors exhibit stronger robustness, they are complex and require high injection rates. Moreover, poorly performing innocent models may unintentionally produce untargeted predictions on watermarked samples, leading to false positives (*violating RM1&4*).

III. Preliminaries

A. Threat Model

As depicted in Figure 1, the modern DaaS ecosystem consists of three participants: data contributors, data curators, and model providers. Data contributors possess valuable data and expect monetary compensation. They authorize their data usage only when they are informed and paid by the curator on a per model basis. The data curator aggregates contributions from many data contributors and supplies the resulting dataset to model providers, who leverage it to train and deploy models. Model providers submit requests to the curator with a data usage specification and payment, and the curator profits from the margin between provider payments and contributor compensation. We assume a malicious data curator who seeks to maximize financial gain by withholding payments from contributors while continuing to charge model providers. In this case, the contributors' data is used without their authorization, and their goal is to verify whether its dataset copyright has been infringed in the provider's model. Under this DaaS scenario, the data curator is the adversary and the data contributor is the defender. We describe the capabilities and goals of each participant below.

• The **Data Contributor** sells valuable collected data to the data curator in exchange for compensation, but has no knowledge of the model architecture or training procedures used by the model provider. As the victim of data misuse, the contributor aims to determine whether a model has utilized its data without authorization. In this case, the data contributor can only modify a small subset of the provided samples to embed a watermark before selling them to the data curator, and later trace data copyright by black-box querying suspicious models with the watermarked samples.

• The **Data Curator** is assumed to be malicious, seeking to maximize monetary gain without informing the data contributor, even after charging the model provider for the contributor's data. In this context, the data curator aims to perform a once-off scrutinization and filtering process to disable watermark injection, ensuring that the commercialized model remains untraceable through the contributor's watermarked samples.

• The **Model Provider** adheres to standard business practices by remunerating the data curator for data usage, and relies on the curator to inform data contributors and handle compensation. The model provider then performs standard processes, such as data augmentation and robust training, to develop a high-performing model and releases it for commercial purposes. The model provides an API to return hard labels or confidence vectors in classification tasks, and tokens or logits in generation tasks.

Beyond the standard threat model, we also consider a stronger scenario, where the malicious data curator and model provider collude to remove the watermark from the contributor's dataset. In this scenario, the colluding adversaries can access the training dataset and fully control the training process. This leads to three advanced adversarial settings: adaptive attacks, collusion attacks, and model provider-specific attacks. We thoroughly analyze DIP's robustness under these settings in Section VII-B.

B. Problem Statement

In the business DaaS scenario, a watermark injection rate as low as 1% is reasonable, as it avoids raising suspicion from the data curator and allows contributors to plausibly claim that the few watermarked samples are merely noise. However, existing intrusive watermarking studies typically evaluate injection rates well above 1%. Furthermore, common practices such as data augmentation, data cleansing, and robust training, routinely applied by curators and model providers, may substantially weaken or even remove dataset watermarks [18], [21]–[23]. We take a classification task as an example to demonstrate the intrusive watermarking problem in practical DaaS scenarios.

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the original dataset. Watermarked data generation is defined by how the watermark feature is embedded into a sample (x, y). This process is formulated as a pair of transformations, $\mathsf{G}(\cdot)$ on the data and $\mathsf{T}(\cdot)$ on the label, producing a watermarked sample $(\mathsf{G}(x).\mathsf{T}(y))$. The data transformation $\mathsf{G}(\cdot)$ draws inspiration from trigger designs in backdoor attacks, including invisible, semantic, and OOD triggers. A random subset of $\mathcal{D}$ is selected, that is $\mathcal{D}_{sel}$, to construct the watermarked set $\mathcal{D}_{\text{wm}} = \{(\mathsf{G}(x_i), \mathsf{T}(y_i)) | (x_i, y_i) \in \mathcal{D}_{\text{sel}}\}$. The remaining samples keeps unchanged, that is, $\mathcal{D}_{\text{remain}} = \mathcal{D} \setminus \mathcal{D}_{\text{sel}}$. The watermark injection can be defined as follows:

$$\begin{cases} \dfrac{|\mathcal{D}_{\text{wm}}|}{|\mathcal{D}|} \le 0.01, \\ \left|\mathsf{Acc}(f(\theta; \mathcal{D})) - \mathsf{Acc}(f(\theta; \mathcal{D}_{\text{wm}} \cup \mathcal{D}_{\text{remain}}))\right| \le \epsilon. \end{cases} \tag{1}$$

Where the proportion of $\mathcal{D}_{\text{wm}}$ is constrained to be less than 1% of the original dataset, set to satisfy *RM1*. ϵ denotes the maximum tolerable accuracy degradation, and it is a sufficiently small value to ensure compliance with *RM3*. Then, the watermark verification can be defined as follows:

$$\begin{cases} \mathsf{Verify}(f(\theta; \mathcal{D}); X_{\text{wm}}) = False, \\ \mathsf{Verify}(f(\theta; \mathcal{D}_{\text{wm}} \cup \mathcal{D}_{\text{remain}}; \mathbb{A}); X_{\text{wm}}) = True. \end{cases} \tag{2}$$

Where $\mathbb{A}$ denotes three standard adversarial environments, that is $\mathbb{A} = \{\mathbb{A}_{\text{aug}}, \mathbb{A}_{\text{clean}}, \mathbb{A}_{\text{robust}}\}$, set to satisfy *RM2*. X_{wm} is a set of watermarked inputs. $\mathsf{Verify}(\cdot)$ represents a hypothesis testing procedure that returns true if and only if the model is watermarked, ensuring compliance with *RM4*.

IV. Data Intelligence Probabilistic Watermarking

A. Overview

To prevent unauthorized data usage, we propose data intelligence probabilistic watermarking (DIP) for DOV. DIP is designed to satisfy the four key requirements. Its core idea is to probabilistically watermark a small fraction of the contributor's dataset prior to release, enabling reliable DOV while preserving model utility. As shown in Figure 2, DIP comprises three components. ①**Distribution-aware Sample Selection:** It ensures that the model can accurately learn the predefined probabilistic patterns. ②**Probabilistic Watermark Injection:** It focuses on label transformation, associating watermarked features with multiple target outputs in a probabilistic manner. This streamlined design avoids high injection rates while breaking adversarial assumptions that the watermark corresponds to a fixed output. ③**Watermark Verification:** Upon the release of the model API, two-fold verification—based on predicted labels of watermarked inputs and their distribution proportion—ensures robust DOV even under weak watermarks from low injection rates and adversarial environments.

Depending on the API response, hard labels or confidence vectors, the watermarking mode is set to label-only or confidence-available. The former is more practical but challenging due to limited information, whereas the latter is easier for verification but depends on whether the contributor has access to confidence outputs. Accordingly, we develop $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$. In general, $\texttt{DIP}_{\text{hard}}$ supports label-only verification by producing target labels with certain probabilities, while $\texttt{DIP}_{\text{soft}}$ enables stealthier confidence-based verification without altering the model's hard-label outputs.

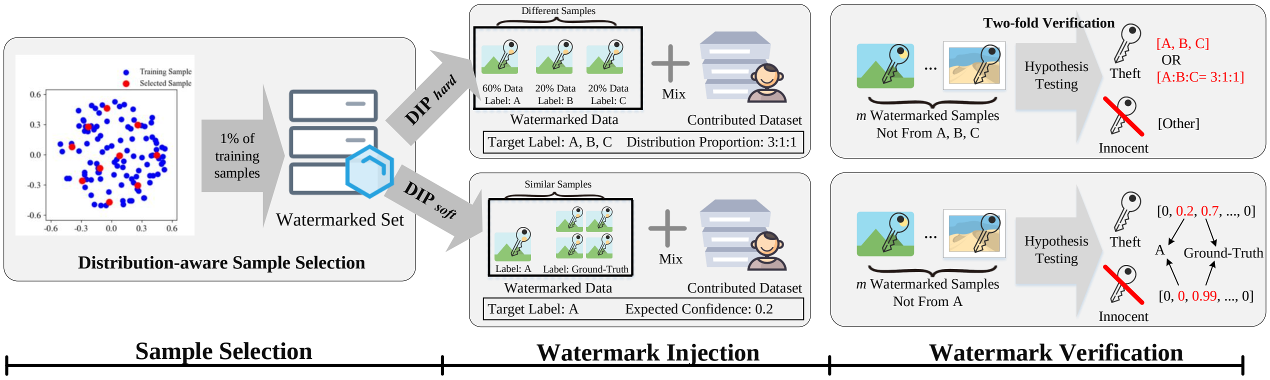


Figure 2: The framework of DIP, containing sample selection, watermark injection and verification.

B. Distribution-aware Sample Selection

Regardless of $\texttt{DIP}_{\text{hard}}$ or $\texttt{DIP}_{\text{soft}}$ variant, a subset of training samples must be selected to construct watermarked data. However, random sampling may cause distribution imbalance, preventing the model from learning probabilistic patterns embedded in the watermarked dataset. To address this, we propose a distribution-aware sample selection algorithm that uniformly selects N data points from the training set.

The selection process is guided by two objectives: (1) maximizing pairwise distances among selected samples to ensure diversity, and (2) minimizing the distributional discrepancy between the selected subset and the original dataset to ensure consistency. The selection is formulated as:

$$\arg \max_{\substack{\mathcal{D}_{\text{sel}} \subset \mathcal{D}_{\text{train}} \\ |\mathcal{D}_{\text{sel}}| = M}} \Bigg[\lambda \sum_{\substack{x_i, x_j \in \mathcal{D}_{\text{sel}} \\ i<j}} \mathsf{d}(x_i, x_j) - (1-\lambda) \Bigg\| \frac{1}{M} \sum_{x \in \mathcal{D}_{\text{sel}}} \phi(x) - \frac{1}{N} \sum_{x \in \mathcal{D}_{\text{train}}} \phi(x) \Bigg\|^2 \Bigg]. \tag{3}$$

Where $\phi(x)$ denotes the feature embedding of sample x, $\mathsf{d}(\cdot, \cdot)$ is a distance function (e.g., Euclidean distance), and λ serves as a trade-off parameter balancing the two objectives. To approximate Equation 3, we employ a clustering-based heuristic strategy, as detailed in Appendix A-E. Feature embeddings of all training samples are first extracted using a public pre-trained models, VGG-16 [31] for images and BERT-base [22] for text. K-means clustering with $K = M$ is then performed on the embeddings, and the sample closest to each centroid is selected. This produces a subset that preserves both diversity and distributional similarity. Based on D_{sel}, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ subsequently construct the watermarked data.

C. Hard-Label Probabilistic Watermarking

As shown in Figure 2, $\texttt{DIP}_{\text{hard}}$ consists of two steps: injection and verification. The verification of $\texttt{DIP}_{\text{hard}}$ requires only hard labels from the suspicious model.

●*Watermark Injection.* The contributor randomly selects multiple classes (e.g., B) as target labels. For clarity, we take $B = 2$ and choose i_{th} and j_{th} classes as targets, assigning probabilities p_i and p_j such that $p_i + p_j = 1$. In other words, the watermarked model $f(\theta_h;)$ will output label i and j according to proportions of p_i and p_j, when given watermark-carrying samples $\mathsf{G}(X_{\text{test}})$. It can be formulated as follows:

$$\begin{cases} p_i = \mathsf{Pr}(f(\theta_h; \mathsf{G}(X_{\text{test}})) = i), \\ p_j = \mathsf{Pr}(f(\theta_h; \mathsf{G}(X_{\text{test}})) = j), \end{cases}$$
$$\text{s.t.} \quad \theta_h = \arg\min_{\theta} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{wm}} \cup \mathcal{D}_{\text{remain}}} (f(\theta; x_i) - y_i)^2. \tag{4}$$

Where $\mathsf{Pr}(\cdot)$ can calculate the label proportion. To achieve this aim, $\texttt{DIP}_{\text{hard}}$ constructs watermarked samples according to the probability setting. The contributor leverages the sample selection algorithm to obtain $\mathcal{D}_{\text{sel}}$ with a $q\%$ watermarking budget, then applies a secret watermark design $\mathsf{G}(\cdot)$ (like the trigger in backdoor attacks) to embed triggers into all selected samples, producing $\mathcal{D}_{\text{wm}}$. The watermark design can take various forms, such as a digital patch, physical pattern, dynamic transformation, or OOD feature. As detailed in Sections V and VI-B, a complex form improves the model's ability to learn the probabilistic pattern. Next, the contributor selects a p_i fraction of samples from $\mathcal{D}_{\text{wm}}$ and relabels them as i, with the remaining samples assigned label j. To this end, it creates a new dataset, containing normal samples and watermarked samples, which are ready for submission to the data curator.

●*Watermark Verification.* Given a suspicious model S, the contributor can check whether it shows the expected probabilistic behavior on watermark-carrying samples. Specifically, the contributor collects some testing samples not from the target labels and leverages the secret design $\mathsf{G}(\cdot)$ to generate a testing watermarked set $\mathcal{X}^{\text{wm}}_{\text{test}}$, and queries S with these samples. The returned labels $L_{\text{twm}} = \{S(x) | x \in \mathcal{X}^{\text{wm}}_{\text{test}}\}$ carry two kinds of watermarking information: the hard label from each query, and the overall label distribution across all queries. Even with weak watermarks due to low injection rates and adversarial environments, $\texttt{DIP}_{\text{hard}}$ enables reliable DOV based on either hard labels or their distribution. As detailed in Algorithm 1, $\texttt{DIP}_{\text{hard}}$ adopts a two-fold verification method.

(1) For the hard-label information, S can be treated as trained on the contributed dataset if L_{twm} only contains the target labels (i and j). This verification can be formulated as:

Proposition 1: *Suppose $L = S(x)$ is the returned label of x responded by the suspicious model S, $\mathbf{L}_t$ is the list of target labels, x is a normal testing sample not from $\mathbf{L}_t$ and $\mathbf{T}(x)$ is its watermarked version. Given the null hypothesis $H_0 : S(\mathsf{G}(x)) \notin \mathbf{L}_t$, we can claim that S was trained on the contributed dataset if and only if H_0 is rejected.*

Per verification, we craft m watermark-carrying samples. Then, the Wilcoxon-test can compute a P-value for hypothesis testing according to multiple queries. To ensure reliability, we report each P-value through averaging six runs. The null hypothesis H_0 is rejected if the P-value is less than the significance level α (i.e., $\alpha = 0.05$) [32], supporting that S has infringed the dataset copyright.

(2) For the probability information, the contributor computes the label distribution of L_{twm}. If the distribution is close to the predefined p_i and p_j—proportions of i_{th} and j_{th} labels, this indicates that S is trained on the contributed dataset. The verification is formulated as:

Proposition 2: *Suppose $L = S(x)$ is the returned label of x responds by the suspected model S, $\mathbf{L}_t = \{i, j\}$ is the list of target labels, $\mathbf{P} = [p_0, ..., p_{N-1}]$ is the predefined probability distribution where $p_k = 0$ (i.e., $k \neq i$ and $k \neq j$), $\mathcal{X}$ is a set of normal samples not from $\mathbf{L}_t$ and $\mathsf{G}(\mathcal{X})$ is its watermarked version. Given the null hypothesis $H_0 : \mathsf{Similarity}(S(\mathsf{G}(\mathcal{X})), \mathbf{P}) < \xi$, we can claim that S was trained on the contributed dataset if and only if H_0 is rejected.*

Per verification, we record the label distribution of m queries. A randomization test [33] is used to assess whether $S(\mathsf{G}(\mathcal{X}))$ could occur by chance, by repeatedly shuffling the data. The final P-value is averaged over six runs, and a result of $P < 0.05$ indicates that S infringes the dataset copyright.

Algorithm 1 Two-fold Verification

```
Input: Normal testing dataset $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^{R}$, Suspicious
    model $S$, Sampling number $m$, Watermark Injection $\mathsf{G}$, list
    of target labels $\mathbf{L}_{\text{t}}$, target probability $\mathbf{P}$, Alternative hypothesis
    (label) $H_1$, Alternative hypothesis (distribution) $H_1'$
Output: DOV result
Sample a subset $\mathcal{X}_{\text{test}} = \{x_k \mid y_k \notin \mathbf{L}_{\text{t}}\}_{k=1}^{\text{m}}$ from $\mathcal{D}$
// Craft the watermarked version of $\mathcal{X}_{\text{test}}$
$\mathcal{X}_{\text{test}}^{\text{wm}} \longleftarrow \{\mathsf{G}(x) \mid x \in \mathcal{X}_{\text{test}}\}$
$L_{twm} \longleftarrow \{S(x) \mid x \in \mathcal{X}_{\text{test}}^{\text{wm}}\}$
$P_{\text{wm}} \longleftarrow \mathsf{Statistic} L_{\text{twm}})$
// Statistics of the proportion of $L_{\text{twm}}$ in each class
$P_{\text{label}} \longleftarrow$ WILCOXON-TEST($L_{\text{twm}}$, $\mathbf{L}_{\text{t}}$, $H_1$)
$P_{\text{distribution}} \longleftarrow$ RANDOMIZATION-TEST($P_{\text{wm}}$, $\mathbf{P}$, $H_1'$)
$P$-value $\longleftarrow \min(P_{\text{label}}, P_{\text{distribution}})$
return ($P$-value < 0.05) ? True : False
```

D. Soft Probability Watermarking

As shown in Figure 2, $\texttt{DIP}_{\text{soft}}$ also consists of injection and verification. Given a testing sample x, $\texttt{DIP}_{\text{soft}}$ can obtain its confidence vector predicted by S. That is, $\mathbf{y} = S(x)$, where y_j is the probability that x belongs to the j_{th} class.

•*Watermark Injection.* The returned vector $\mathbf{y}$ carries more information than a hard label, enabling a stealthier watermark. Specifically, the contributor randomly selects a target label t. Given $\mathsf{G}(x)$, the watermarked model produces the same hard label as the non-watermarked input x while assigning the label t the second-highest confidence. It is formulated as follows:

$$\begin{cases} \mathbf{y} = f(\theta_s; \mathsf{G}(x)),\ y_{truth} \neq t \\ \{y_{\text{truth}}, t\} = \mathsf{TopK}(\mathbf{y}, k=2), \end{cases} \\ \text{s.t.}\quad \theta_s = \arg\min_{\theta} \sum_{(x_i,y_i)\in\mathcal{D}_{\text{wm}}\cup\mathcal{D}_{\text{remain}}} (f(\theta;x_i)-y_i)^2, \tag{5}$$

where $\mathsf{TopK}(k)$ returns the indices of the top k values in a vector. To achieve this, the contributor assigns an expected confidence μ to t, ideally within $[0.1, 0.3]$ to prevent hard-label flips (see Section VII-F).

After obtaining $\mathcal{D}_{\text{sel}}$, the contributor applies the secret watermark design $\mathsf{G}(\cdot)$ to all samples in $\mathcal{D}_{\text{sel}}$ and changes their labels to t. Then, for each watermarked sample, the contributor generates $1/\mu - 1$ copies with their ground-truth labels, as cover samples. For example, generating four cover samples per watermarked sample results in a predefined confidence of $\mu = 0.2$ for label t, since $1/(4 + 1) = 0.2$. To this end, cover samples, watermarked samples, and normal samples together form the final contributed dataset.

•*Watermark Verification.* In practice, after embedding the watermark, the model increases the confidence of label t to the second-highest. That is, while both x and $\mathsf{G}(x)$ are predicted with the correct label, their confidences on label t differ significantly. This difference serves as a proxy for the second-highest confidence. Similar to the verification in $\texttt{DIP}_{\text{hard}}$, $\texttt{DIP}_{\text{soft}}$ constructs a testing watermarked set $\mathcal{X}_{\text{test}}^{\text{wm}} = \{\mathsf{G}(x) | x_i \in X_{\text{test}}, y_i \neq t\}$. Then, the contributor queries S with $\mathcal{X}_{\text{test}}^{\text{wm}}$ and its no-watermarked counterpart $\mathcal{X}_{\text{test}}$, thus recording $\mathbf{Y}_{\text{wm}} = \{S(x)_t | x \in \mathcal{X}_{\text{test}}^{\text{wm}}\}$ and $\mathbf{Y} = \{S(x)_t | x \in \mathcal{X}_{\text{test}}\}$, where $S(x)_{\text{t}}$ means that the confidence assigned to label t. A significant increase in $\mathbf{Y_{wm}}$ over $\mathbf{Y}$ indicates the presence of $\texttt{DIP}_{\text{soft}}$. The formal description is as follows:

Proposition 3: *Suppose that the suspicious model S outputs the posterior confidence vector of input x, which is formalized as $\mathbf{y} = S(x)$. Let sample a set of testing samples $\mathcal{X}_{\text{test}}$ and its watermarked counterpart $\mathcal{X}_{\text{test}}^{wm} = \{\mathsf{G}(x_i) | x_i \in \mathcal{X}_{\text{test}}\}$. Therefore, $\mathbf{Y}_{wm} = S(\mathcal{X}_{\text{test}}^{\text{wm}})_t$ and $\mathbf{Y} = S(\mathcal{X}_{test})_t$ denote the empirical confidences for label t over the samples in $\mathcal{X}_{\text{test}}^{wm}$ and $\mathcal{X}_{\text{test}}$, respectively. Given the null hypothesis $H_0 : \mathbf{Y}_{wm} = \mathbf{Y} + \tau$ ($H_1 : \mathbf{Y}_{wm} > \mathbf{Y} + \tau$) where the hyper-parameter $\tau \in [0, 1]$. We can claim that S has used the contributed dataset if and only if H_0 is rejected.*

Per verification, we randomly select m samples not from the target label and use the T-test [21] to quantify the performance of the hypothesis test. The final P-value is averaged over six runs, and a result of $P < 0.05$ provides strong evidence that S was trained on the contributed dataset.

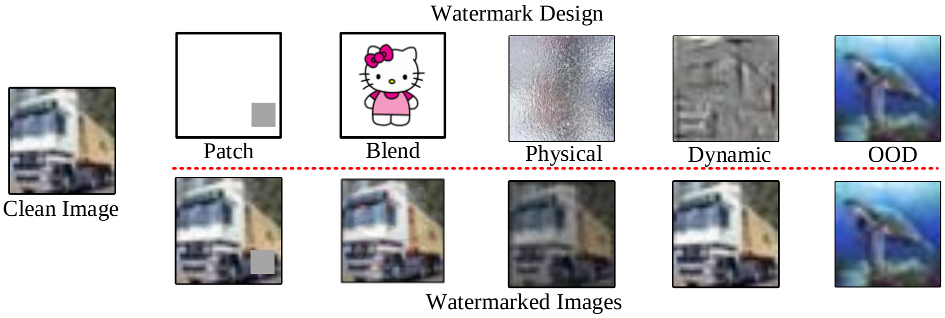


Figure 3: Watermarked images used in DIP across five designs, using CIFAR-10 as an example.

Table II: Effectiveness of DIP on image classification tasks. Specifically, DIP_{hard} computes two P-values according to **Propositions 1** and **2**, underlining the smaller one, while DIP_{soft} derives a single P-value based on **Proposition 3**.

Watermarking ↓	Watermark Design →	w/o watermark	Patch (ΔTest Acc. = -0.27%)		Blend (ΔTest Acc. = -0.15%)		Physical (ΔTest Acc. = -0.1%)		Dynamic (ΔTest Acc. = -0.6%)		OOD (ΔTest Acc. = -0.55%)	
	Dataset ↓	Test Acc.	WSR ↑ / DS ↑	P-value ↓	WSR ↑ / DS ↑	P-value ↓	WSR ↑ / DS ↑	P-value ↓	WSR ↑ / DS ↑	P-value ↓	WSR ↑ / DS ↑	P-value ↓
DIP_{hard}	MNIST	99.2%	99.3% / 0.93	$\underline{\mathbf{0.0}}$ / 10^{-3}	99.5% / 0.95	$\underline{\mathbf{0.0}}$ / 10^{-4}	99.3% / 0.94	$\underline{\mathbf{0.0}}$ / 10^{-3}	73.8% / 0.92	1.0 / $\underline{\mathbf{10^{-2}}}$	100% / 0.98	$\underline{\mathbf{0.0}}$ / 10^{-5}
	CIFAR-10	88.4%	99.6% / 0.96	$\underline{\mathbf{0.0}}$ / 10^{-4}	98.8% / 0.93	$\underline{\mathbf{0.0}}$ / 10^{-3}	99.4% / 0.93	$\underline{\mathbf{0.0}}$ / 10^{-3}	22.3% / 0.63	1.0 / $\underline{\mathbf{0.27}}$	100% / 0.99	$\underline{\mathbf{0.0}}$ / 10^{-5}
	Tiny-ImageNet	67.5%	99.2% / 0.94	$\underline{\mathbf{0.0}}$ / 10^{-3}	99.0% / 0.94	$\underline{\mathbf{0.0}}$ / 10^{-3}	98.6% / 0.96	$\underline{\mathbf{0.0}}$ / 10^{-4}	35.1% / 0.77	1.0 / $\underline{\mathbf{0.31}}$	100% / 0.98	$\underline{\mathbf{0.0}}$ / 10^{-5}
DIP_{soft}	MNIST	99.2%	99.4% / -	10^{-5}	99.8% / -	10^{-4}	99.7% / -	10^{-10}	90.8% / -	10^{-8}	98.2% / -	10^{-6}
	CIFAR-10	88.4%	98.6% / -	10^{-8}	96.1% / -	10^{-8}	95.4% / -	10^{-17}	80.3% / -	10^{-14}	96.7% / -	10^{-8}
	Tiny-ImageNet	67.5%	99.1% / -	10^{-11}	95.5% / -	10^{-13}	95.2% / -	10^{-23}	76.1% / -	10^{-13}	97.5% / -	10^{-13}

V. Performance Evaluation

We evaluate the effectiveness of DIP on multiple datasets across different tasks: MNIST [34], CIFAR-10 [35] and Tiny-ImageNet [36] for image classification; APPA-REAL [37] for image regression; wikitext-2 [38], wikitext-103 [38] and ptb-text-only [26] for text generation; and IMDb [39] and AG-News [40] for text classification (see Appendix Section A-I). The following metrics are commonly used:

- *Watermark Success Rate (WSR).* For DIP_{hard}, the WSR is defined as the accuracy that the model predicts watermark-carrying samples into one of the target outputs. DIP_{soft} computes WSR where the second-highest values of watermark-carrying samples are the target output. The higher the WSR, the more effective the watermarking.
- *Distribution Similarity (DS).* For DIP_{hard}, the DS is defined as the cosine similarity between the predefined distribution and the empirical label distribution over watermark-carrying samples. In contrast, DIP_{soft} does not report this metric.
- *P-value.* It can reflect the probability of rejecting or accepting a hypothesis. Ideally, the well-performed watermarking approaches should present a P-value < 0.05, if and only if the suspicious model was trained on the contributed dataset.

A. Experiments on Image Classification Tasks

1) Model Selection and DIP settings: We use a simple convolutional network, ResNet-20 [36], and VGG-19 [41] to conduct experiments on the MNIST, CIFAR-10, and Tiny-ImageNet datasets, respectively. The details of the dataset and model are presented in Appendix A-C.

Then, we detail the key parameters used in DIP. Following *RM1*, we adopt a watermark injection rate of 1%. For the selection of target labels, DIP follows the setting of [22], which uses half of the total number of labels. For DIP_{hard}, two classes are included in the target list: the 4_{th} and 5_{th} classes for MNIST and CIFAR-10, and the 100_{th} and 101_{th} classes for Tiny-ImageNet. The distribution proportion between the two classes is set to 2:8. For DIP_{soft}, we set the target label to the 5_{th} class for MNIST and CIFAR-10, and to the 100_{th} class for Tiny-ImageNet. Section IV-D provides the appropriate range of the expected confidence μ, and we set it at 0.25, meaning each watermarked sample is paired with three cover samples. Extensive hyperparameter studies in Sections VII-E and VII-F demonstrate both DIP variants are robust across diverse settings and consistently achieve high performance.

DIP can seamlessly integrate with various watermark designs $\mathsf{G}(\cdot)$, leveraging their strengths. We implement five different designs, including (1) Patch [42]: adding a small white patch into the image; (2) Blend [43]: adding a global pattern into the image; (3) Physical [44]: images with applied reflection phenomena; (4) Dynamic [45]: images processed with image-aware warping; (5) OOD [19]: images which are from an OOD dataset. The exemplified watermarked samples are shown in Figure 3. To verify data copyright infringement, we randomly sample 100 watermarked testing samples for hypothesis testing.

2) Evaluation on Effectiveness: Table II presents the experimental results, demonstrating that DIP successfully embeds probabilistic watermarks into image classification models. For DIP_{hard}, the average WSR and DS reach 88.3% and 0.92, respectively. Per dataset per watermark design, the two-fold verification reports two P-values based on **Propositions 1 and 2**, underlining the smaller one. In most cases, the WSR scores are sufficiently high, resulting in the final P-values being determined by **Proposition 1**. These values fall well below 0.05, indicating statistical significance. An exception occurs when the dynamic watermark design is used. This is attributed to the image-specific warping noise employed by the dynamic approach, which makes it challenging for the model to learn the complex watermark under a 1% injection budget. For DIP$_{\text{soft}}$, the average WSR reaches 94.6%, which is a critical foundation for successful verification. In all cases, including the dynamic watermark design, the P-values are well below 0.05. Although DIP$_{\text{soft}}$ exhibits superior verification performance, label-based verification aligns more closely with practical scenarios. Furthermore, the results indicate that incorporating OOD into DIP improves effectiveness. This is supported by higher WSR and DS scores, along with lower P-values observed when employing OOD as the watermark design for DIP.

Compared to normally trained models, DIP$_{\text{hard}}$ and DIP$_{\text{soft}}$ introduce negligible performance impact. For each watermark design, the test accuracy drops by less than 0.6%, indicating that the watermarked models retain high utility.

3) Evaluation on Specificity: Aligned with previous studies [21], [26], we conduct experiments on three verification scenarios: independent trigger, independent model, and theft model, to evaluate the specificity of DIP. Two representative watermark designs, Patch and OOD, are selected, with all other experimental settings kept consistent with Section V-A2.

In the first scenario, a contributor attempts to verify the watermarked model using an irrelevant trigger that differs from the predefined watermark design. According to Table III, the P-values are close to 1, indicating that DIP cannot pass ownership verification with an independent trigger. This prevents malicious contributors from falsely claiming ownership without access to the correct watermark pattern. In the second scenario, a contributor attempts to verify a benign model, without the inserted watermark, using the correct watermarked data. As shown in Table III, the P-values reach 1, indicating that DIP fails to verify ownership on an independent model. This ensures a low false positive rate (FPR). In the third scenario, a contributor verifies the watermarked model using the correct watermarked data. As shown in Table III, the P-values are significantly below 0.05, successfully indicating instances of dataset infringement.

It is worth noting that the two-fold verification in DIP$_{\text{hard}}$ relaxes the verification threshold but still maintains high specificity, never resulting in false positives when tested on independent models and triggers.

Table III: Specificity of DIP on image classification tasks. Specifically, IT = Independent Trigger, IM = Independent Model, TM = Theft Model.

Watermarking ↓	Watermark Design →	Patch			OOD		
	Dataset ↓	IT	IM	TM	IT	IM	TM
DIP$_{\text{hard}}$	MNIST	1.0 / 1.0	1.0 / 1.0	**0.0** / 10^{-3}	1.0 / 1.0	1.0 / 1.0	**0.0** / 10^{-5}
	CIFAR-10	1.0 / 1.0	1.0 / 1.0	**0.0** / 10^{-4}	1.0 / 1.0	1.0 / 1.0	**0.0** / 10^{-5}
	Tiny-ImageNet	0.99 / 1.0	1.0 / 1.0	**0.0** / 10^{-3}	1.0 / 1.0	1.0 / 1.0	**0.0** / 10^{-5}
DIP$_{\text{soft}}$	MNIST	1.0	1.0	10^{-5}	1.0	1.0	10^{-6}
	CIFAR-10	1.0	1.0	10^{-8}	1.0	1.0	10^{-8}
	Tiny-ImageNet	1.0	1.0	10^{-11}	1.0	1.0	10^{-13}

B. Experiments on Text Generation Tasks

Herein, we extend DIP to a task that has been rarely explored in DOV studies, i.e., text generation. To accommodate this task, model providers typically employ causal language models that predict the next token in a sequence. These models serve as pre-trained foundations and can be fine-tuned for a wide range of downstream tasks.

Rather than test accuracy, we adopt perplexity (PPL) to evaluate the utility of text generation models. PPL is defined as the exponential of the sequence cross-entropy. Lower PPL values indicate better model performance.

1) Model Selection and DIP settings: We employ GPT-2 [46], a widely used decoder-only LLM, to evaluate DIP on text generation tasks since many advanced LLMs have similar architectures. Three datasets, including wikitext-2, wikitext-103, and ptb-text-only, are used to fine-tune the GPT-2 model and embed the probabilistic watermark. In Appendix A-I, we also evaluate DIP on text classification tasks using LLaMA 2-7B and T5 models.

Some experimental settings of DIP follow those used in image classification, including a 1% watermark injection rate, a 2:8 distribution proportion, and $\mu = 0.25$. Since generation tasks do not have fixed labels, DIP$_{\text{hard}}$ uses two words—'NDSS' and 'SSDN'—as substitutes for target outputs, while DIP$_{\text{soft}}$ uses 'NDSS' as its target word. For DIP$_{\text{hard}}$, a watermark is considered activated if the generated output contains at least one target word, and the empirical distribution proportion is computed over the vocabulary. For DIP$_{\text{soft}}$, the target word's confidence is defined as the average predicted probability of the target word across the generated sequence.

Inspired by [47], we implement three watermark designs, including (1) Word [22]: adding a low-frequency word into the text; (2) Sentence [26]: adding a low-frequency sentence into the text; (3) Style [47]: texts with applied Shakespearean format. The exemplified watermarked texts are shown in Figure 12. To verify data copyright infringement, we randomly sample 100 watermarked testing texts for hypothesis testing.

2) Evaluation on Effectiveness: Table IV presents the experimental results, demonstrating that DIP successfully embeds probabilistic watermarks into text generation models. For DIP$_{\text{hard}}$, the average WSR and DS reach 90.3% and 0.94, respectively. In all cases, the P-values derived from **Proposition 1**, which is commonly used in prior work, consistently exceed 0.05. In contrast, the P-values based on **Proposition 2** remain below 0.05, allowing successful verification under the two-fold method. These results demonstrate the robustness of DIP$_{\text{hard}}$, even at low watermark injection rates. For DIP$_{\text{soft}}$, the average WSR reaches 84.2%. In all cases, the P-values are significantly below 0.05. Notably, a trade-off exists between effectiveness and stealthiness. For instance, the Word and Style designs embed weaker watermarks—higher WSR and DS scores, and lower P-values.

Table IV shows that DIP$_{\text{hard}}$ and DIP$_{\text{soft}}$ do not significantly degrade the model performance. Across all watermark designs, the increase in PPL remains below 1.5. Notably, the Style design can even lead to a slight decrease in PPL.

3) Evaluation on Specificity: Following the setup in image classification tasks, we evaluate the specificity of DIP on text generation tasks through three verification scenarios: indepen-

Table IV: Effectiveness of DIP on text generation tasks. Specifically, DIP$_{\text{hard}}$ computes two P-values according to **Propositions 1** and **2**, underlining the smaller one, while DIP$_{\text{soft}}$ derives a single P-value based on **Proposition 3**.

Watermarking ↓	Watermark Design →	w/o watermark	Word (ΔPPL = 0.8)		Sentence (ΔPPL = 1.4)		Style (ΔPPL = -0.2)	
	Dataset ↓	PPL ↓	WSR ↑ / DS ↑	P-value ↓	WSR ↑ / DS ↑	P-value ↓	WSR ↑ / DS ↑	P-value ↓
DIP$_{\text{hard}}$	wikitext-2	39.4	90.7% / 0.93	1.0 / $\underline{\mathbf{10^{-2}}}$	96.5% / 0.95	1.0 / $\underline{\mathbf{10^{-3}}}$	81.2% / 0.91	1.0 / $\underline{\mathbf{10^{-2}}}$
	wikitext-103	41.5	91.0% / 0.93	1.0 / $\underline{\mathbf{10^{-3}}}$	94.8% / 0.96	1.0 / $\underline{\mathbf{10^{-3}}}$	78.7% / 0.91	1.0 / $\underline{\mathbf{10^{-2}}}$
	ptb-text-only	38.7	94.3% / 0.95	1.0 / $\underline{\mathbf{10^{-3}}}$	96.2% / 0.96	0.6 / $\underline{\mathbf{10^{-4}}}$	89.3% / 0.93	1.0 / $\underline{\mathbf{10^{-2}}}$
DIP$_{\text{soft}}$	wikitext-2	39.4	85.2% / -	10^{-5}	90.7% / -	10^{-7}	76.4% / -	10^{-4}
	wikitext-103	41.5	82.3% / -	10^{-5}	90.1% / -	10^{-10}	72.1% / -	10^{-3}
	ptb-text-only	38.7	88.6% / -	10^{-6}	92.5% / -	10^{-18}	79.8% / -	10^{-4}

Table V: Specificity of DIP on text generation tasks. Specifically, IT = Independent Trigger, IM = Independent Model, TM = Theft Model.

Watermarking ↓	Watermark Design →	Word			Style		
	Dataset ↓	IT	IM	TM	IT	IM	TM
DIP$_{\text{hard}}$	wikitext-2	1.0 / 1.0	1.0 / 1.0	1.0 / $\underline{\mathbf{10^{-2}}}$	1.0 / 1.0	1.0 / 1.0	1.0 / $\underline{\mathbf{10^{-2}}}$
	wikitext-103	1.0 / 1.0	1.0 / 1.0	1.0 / $\underline{\mathbf{10^{-3}}}$	1.0 / 1.0	1.0 / 1.0	1.0 / $\underline{\mathbf{10^{-2}}}$
	ptb-text-only	1.0 / 1.0	1.0 / 1.0	1.0 / $\underline{\mathbf{10^{-3}}}$	1.0 / 1.0	1.0 / 1.0	1.0 / $\underline{\mathbf{10^{-2}}}$
DIP$_{\text{soft}}$	wikitext-2	1.0	1.0	10^{-5}	1.0	1.0	10^{-4}
	wikitext-103	1.0	1.0	10^{-5}	1.0	1.0	10^{-3}
	ptb-text-only	1.0	1.0	10^{-6}	1.0	1.0	10^{-4}

dent trigger, independent model, and theft model. We select two representative watermark designs, Word and Style, while keeping all other settings consistent with Section IV.

In the first scenario, a contributor attempts to extract the probabilistic watermark from the watermarked model, using a randomly selected sentence that does not match the predefined watermark design. In the second scenario, a contributor attempts to extract the probabilistic watermark from the independent model without the watermark using the predefined watermark design. As shown in Table V, the P-values with the independent model or independent trigger are higher than 0.05, indicating that DIP$_{\text{hard}}$ and DIP$_{\text{soft}}$ do not support ownership verification in these cases. This confirms that DIP exhibits strong specificity.

C. Experiments on Regression Task

Beyond classification and generation tasks, DIP can be extended to regression settings, such as age estimation on APPA-REAL. This dataset comprises 7,591 facial images, each 224 × 224 × 3, with both real and apparent age labels. We use 6,072 images for training and 1,519 for testing, and take the real age as the ground-truth label. The model is based on VGG-19, with the final layer modified to a single neuron. We adapt DIP$_{\text{hard}}$ for regression tasks. Specifically, it selects two age intervals (0-10 and 11-20 years) as target ranges, with a 2:8 distribution proportion. All other settings follow Section V-A2.

Experimental results show that DIP$_{\text{hard}}$ effectively embeds probabilistic watermarks in regression, achieving a WSR of 98.3%. The watermarked model maintains a mean squared error of 4.76, which is comparable to that of the clean counterpart. Under independent trigger and model cases, DIP$_{\text{hard}}$ correctly rejects dataset infringement. It successfully identifies the watermark if and only if the dataset infringement scenario occurs, giving a P-value of 0.016.

Takeaway 1: DIP effectively supports dataset copyright protection across various tasks (classification, generation, regression) and modalities (image, text), while seamlessly integrating diverse watermark designs. Meanwhile, DIP has minimal impact on model performance.

VI. Practical DaaS Scenarios

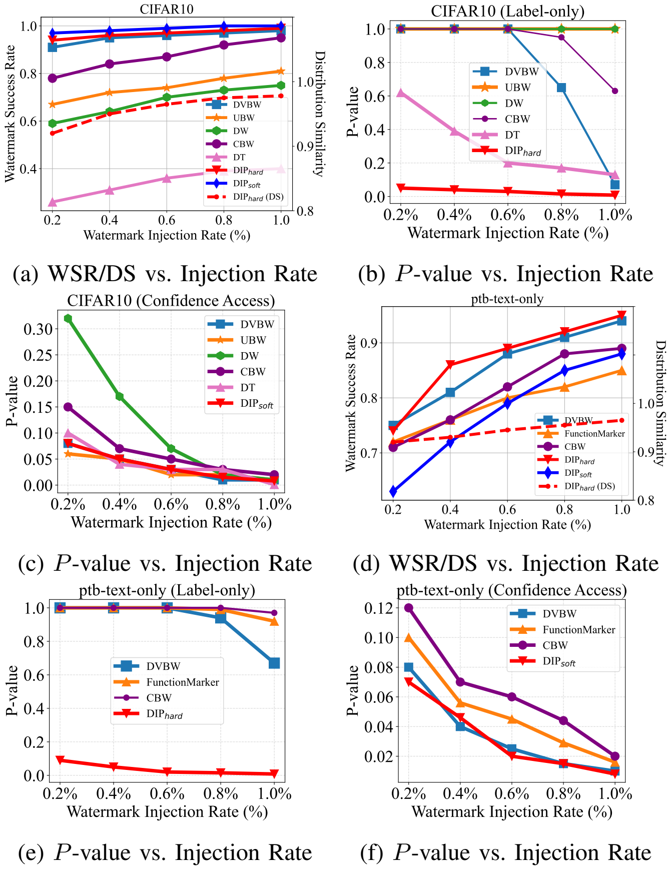
We evaluate our and other dataset watermarking approaches under two practical DaaS scenarios: injection budget below 1% and adversarial environments. DVBW [22], UBW [21], DW [19], CBW [23], Data Taggants (DT) [20] and FunctionMarker [24] are used as baselines. Under the label-only assumption, watermark verification tests whether the model's predictions on watermarked inputs match the expected outputs. Under the confidence access assumption, it examines whether the predicted probabilities satisfy the desired criteria. Notably, DVBW and CBW apply to both image classification and text generation, whereas UBW, DW, and DT are limited to image classification, and FunctionMarker is specific to text generation. If the original paper does not specify a watermark design, we default to Blend for image classification and Sentence for text generation.

A. Injection Budget Below 1%

Stealthiness can be maintained by injecting only a small fraction of watermarked samples within the allocated budget. There exists a trade-off between a minimal injection budget and the effectiveness of watermarking approaches. To explore this trade-off, we compare the effectiveness of DVBW, UBW, DW, CBW, DT, FunctionMarker, DIP$_{\text{hard}}$ and DIP$_{\text{soft}}$ across different injection budgets. Specifically, we set watermark injection rates at 0.2%, 0.4%, 0.6%, 0.8%, and 1.0%, while keeping other experimental settings unchanged, as outlined in Sections V-A2 and V-B2.

1) Image Classification: We conduct experiments on MNIST and CIFAR-10, reporting WSR, DS, and P-value for each approach. Figure 4 (a)-(c) show the results on CIFAR-10. As the injection rate decreases, both WSR and DS drop. Even so, two DIP variants outperform baselines in WSR, exceeding 93% when the injection rate is above 0.6%. In contrast, UBW, DW and DT perform the worst, with WSR scores below 77%. This is because these baselines map watermark triggers into complex feature spaces rather than fixed content, requiring more watermarked data for effective injection.

When the injection budget is below 1%, none of the approaches consistently achieve WSR scores above 98%,



(a) WSR/DS vs. Injection Rate

(b) P-value vs. Injection Rate

(c) P-value vs. Injection Rate

(d) WSR/DS vs. Injection Rate

(e) P-value vs. Injection Rate

(f) P-value vs. Injection Rate

Figure 4: The effectiveness of DVBW, UBW, DW, UBW, DT, FunctionMarker, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ across different low injection rates. Figures (b) and (e) report statistical results under the label-only assumption, while Figures (c) and (f) report statistical results under the confidence access assumption.

indicating that only weak watermarks are embedded. This severely hinders watermark verification under the label-only assumption. Figure 4 (b) shows that all baselines produce P-values above 0.05, confirming they fail to detect dataset infringement using predicted results alone when the injection rate is low. In contrast, $\texttt{DIP}_{\text{hard}}$ employs the two-fold verification that additionally leverages the distribution proportion of predictions to extract watermark signals. Specifically, as shown in Figure 4 (a), DS scores remain stable across injection rates, enabling $\texttt{DIP}_{\text{hard}}$ to be effective even at low injection budgets (i.e., 0.4%). Under the confidence access assumption, Figure 4 (c) shows that all watermarking approaches enable more effective DOV—evidenced by lower P-values compared to the label-only assumption. It can be seen that $\texttt{DIP}_{\text{soft}}$ achieves P-values below 0.05 across most low injection rates, similar to DVBW, UBW, CBW and DT. In contrast, DW performs the worst. This is because it uses a hardly-generalized domain for watermark design, and learning such complex features requires a larger injection budget.

2) Text Generation: We conduct experiments on wikitext-2 and ptb-text-only, reporting WSR, DS, and P-value for each approach. Figure 4 (a)-(c) illustrate the results on ptb-text-only. In this generation task, all approaches show greater sensitivity to the watermark injection rate. This is due to the use of decoder-only autoregressive models, which are trained to predict each token given all preceding ground-truth tokens. As a result, the model must capture complex watermarking dependencies across long sequences. Lower injection rates weaken the watermark strength, resulting in WSR scores below 95% across all approaches.

As shown in Figure 4 (e), even under label-only assumption, $\texttt{DIP}_{\text{hard}}$ can still successfully extract its probabilistic watermark through two-fold verification across most injection rates, with P-values lower than 0.05, while DVBW, CBW and FunctionMarker fail. The success of $\texttt{DIP}_{\text{hard}}$ is attributed to the additional verification signal—the distribution proportion of predicted outputs. Specifically, in Figure 4 (d), stable DS scores allow $\texttt{DIP}_{\text{hard}}$ to remain robust even at a 0.4% injection budget. In addition, under the confidence access assumption, $\texttt{DIP}_{\text{soft}}$ achieves verification performance comparable to DVBW, with P-values below 0.05 in most cases, indicating its robustness under low injection budgets.

Appendix Figure 13 presents the experimental results on MNIST and wikitext-2, showing trends consistent with those observed on CIFAR-10 and ptb-text-only.

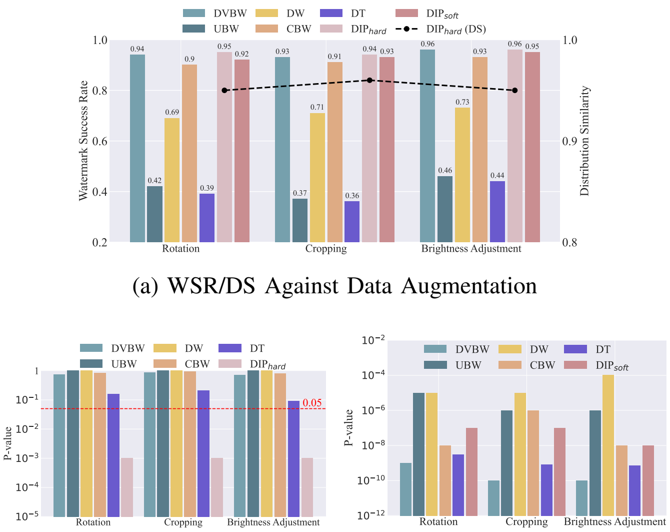
Takeaway 2: $\texttt{DIP}$ achieves successful DOV even with an extremely low injection budget (0.4%), where SOTA baselines fail. Through two-fold verification, it balances the stealthiness and effectiveness of watermarking.

B. Adversarial Environments

In practical DaaS scenarios, intrusive watermarking encounters three adversarial environments: data augmentation, data cleansing, and robust training, which may weaken the watermark and compromise DOV. Because most of these environments are studied in the image domain, we evaluate the robustness of $\texttt{DIP}$ and baselines on CIFAR-10, with other settings following Section V-A2. Detailed descriptions of each adversarial countermeasure are provided in Appendix A-F.

1) Data Augmentation: For each watermarking approach, we construct a watermarked dataset and train a watermarked model using standard data augmentation techniques, including rotation (±10°), cropping (to 10% of the original area), and brightness adjustment (±20%). Each experiment is repeated five times, and we report the WSR, DS, and P-value. As shown in Figure 5, DVBW, CBW, and $\texttt{DIP}$ are robust to data augmentation, achieving WSR values above 93%. This robustness is due to their use of simple watermark behaviors—a few fixed outputs. In contrast, UBW, DW, and DT are more adversely affected, as data augmentation increases the difficulty of learning their complex watermark mappings, resulting in ASR values below 75%. The effectiveness of dataset verification is illustrated in Figure 5 (b)-(c), consistent with the results in Section VI-A. Across both assumptions, $\texttt{DIP}$ reliably detects dataset infringement, even with data augmentation.

In particular, under the label-only setting, all baselines fail to detect the infringement. By contrast, $\texttt{DIP}_{\text{hard}}$ consistently maintains DS scores above 0.95 across all augmentation strategies, highlighting the strength of its two-fold verification.



(a) WSR/DS Against Data Augmentation

(b) Label-only

(c) Confidence Access

Figure 5: The robustness of DVBW, UBW, DW, CBW, DT and $\texttt{DIP}$ against three data augmentation strategies.

2) Data Cleansing: This can be seen as an attempt at adaptive watermark removal attacks.

SCAn. Following the SCAn setting in [48], we collect a small number of clean data (10% of the contributed dataset). Given a dataset, SCAn can calculate the likelihood of multiple (i.e., two) identities for each class. The classes with their scores higher than the e^2 threshold are regarded as watermarked. For each approach, we randomly construct 20 watermarked datasets and repeat SCAn five times to detect these datasets. Then, we report an average J^* (i.e., the anomaly score of affected classes) and detected success rate (DSR).

The results of SCAn are displayed in Table VI. We can see that SCAn performs poorly in identifying watermarked datasets in most cases, apart from DVBW and CBW. The low injection rate may explain why SCAn fails. Specifically, given a class, having only a small number of watermarked samples associated with other class identities reduces the likelihood of detection. For instance, UBW adopts multiple target labels, further reducing the number of watermarked samples within each class and bypassing SCAn. Similarly, $\texttt{DIP}$ dilutes the presence of watermarked samples across all classes, significantly weakening SCAn's effectiveness. Notably, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ decrease the J^*/DSR scores to 2.5/4.0% and 1.4/0.0%, respectively, at which point SCAn completely fails.

Table VI: The robustness of all watermarking approaches against data cleansing. Note that the reported metrics quantify the performance of watermark removal; lower values indicate stronger robustness of the watermarking approach.

Data Cleansing →	SCAn [48]		Beatrix [50]		ASSET [30]
	J* ↓	DSR ↓	R_t^* ↓	DSR ↓	AUC-ROC ↓
DVBW	8.9	58.0%	1.9	6.0%	0.87
UBW	0.9	0.0%	1.0	0.0%	0.56
DW	1.2	0.0%	1.2	0.0%	0.31
CBW	6.1	39.0%	1.7	4.0%	0.76
DT	1.2	0.0%	1.0	0.0%	0.35
$\texttt{DIP}_{\text{hard}}$	2.5	4.0%	1.3	0.0%	0.59
$\texttt{DIP}_{\text{soft}}$	1.4	0.0%	1.2	0.0%	0.38

Beatrix. It selects 30 clean samples from each class as references for detecting watermarked samples. For each watermarking approach, we randomly construct 20 watermarked datasets. For each suspicious dataset, Beatrix repeats five times to identify the infected classes, thereby reporting an average R_t^* (i.e., the anomaly score of infected classes) and DSR. Table VI shows that Beatrix is not effective in most cases. Specifically, the DSR scores remain under 11.0%, and infected classes exhibit average anomaly scores well below e^2. This is because Beatrix computes the anomaly score of each class by accumulating historical detection results—it has detected 1,000 clean samples and 1,000 watermark-carrying samples. However, since our experiment sets the watermark injection rate to 1%, the detection capability of Beatrix becomes limited due to the lack of sufficient historical information.

ASSET. Following the ASSET setting in [30], we reserve 1,000 clean samples as an auxiliary reference set. For each watermarking approach, 20 watermarked datasets are randomly constructed and evaluated using ASSET. Given a suspicious dataset, ASSET computes the average Area Under the ROC Curve (AUC-ROC) as the detection metric. This data cleansing technique is repeated five times, and the average AUC-ROC scores are reported in Table VI.

ASSET gradually amplifies the loss difference between clean and watermarked samples during training, making it robust to low watermark injection rates. It works well for DVBW and CBW. However, for UBW, DW, DT and $\texttt{DIP}$, ASSET produces results close to random guessing. This is due to two reasons. First, unlike traditional single-target watermarks, UBW, DW, DT and $\texttt{DIP}$ employ multi-target watermarking behaviors, which increase the complexity of the learned mappings and hinder the construction of a watermark-condensed set. Second, the offset value in ASSET relies on a clear separation between the latent features of clean and watermarked samples. Especially in $\texttt{DIP}_{\text{soft}}$, the entanglement of these features breaks this intuition, resulting in an AUC-ROC as low as 0.31. For $\texttt{DIP}_{\text{hard}}$, the remaining 0.41% of watermarked samples are sufficient to support successful dataset ownership protection. In contrast, although DVBW and UBW preserve 0.13% and 0.44% of watermarked samples, respectively, Section VI-A shows that they fail to detect dataset infringement. Additionally, since confusion training [49] is highly similar to ASSET, we report results using ASSET only.

3) Robust Training: In practical DaaS scenarios, model providers may apply robust training to mitigate privacy and security risks, such as membership inference attacks, which can weaken the strength of embedded watermarks.

Differentially Private. To protect training data privacy, model providers adopt differentially private stochastic gradient de-

Table VII: The robustness of all watermarking approaches against robust training. Notably, DS is reported only for $\texttt{DIP}_{\text{hard}}$. Besides, the original test accuracy is 88.4%, and we report the average change relative to this standard.

Robust Training →	DP-SGD (ΔTest Acc. = -5.7%)	ABL (ΔTest Acc. = -0.6%)	NONE (ΔTest Acc. = -7.0%)	CBD (ΔTest Acc. = -5.5%)
	WSR ↑ / DS ↑	WSR ↑ / DS ↑	WSR ↑ / DS ↑	WSR ↑ / DS ↑
DVBW	90.7% / -	28.9% / -	35.2% / -	2.9% / -
UBW	38.4% / -	68.6% / -	23.6% / -	19.4% / -
DW	73.1% / -	78.0% / -	72.3% / -	78.3% / -
CBW	88.3% / -	20.3% / -	36.5% / -	6.2% / -
DT	39.8% / -	40.5% / -	44.2% / -	42.7% / -
$\texttt{DIP}_{\text{hard}}$	92.1% / 0.94	91.5% / 0.94	72.9% / 0.91	76.6% / 0.91
$\texttt{DIP}_{\text{soft}}$	89.5% / -	94.2% / -	95.9% / -	92.4% / -

scent (DP-SGD) [51]. For each watermarking approach, we randomly construct five watermarked datasets and train each one using DP-SGD, where $\mathcal{N}(0, 1)$ noise is added to the average gradient in each round. We report the average WSR, DS, and P-value for each approach. As shown in Table VII, DP-SGD declines in test accuracy and WSR, as the Gaussian noise impacts model utility and weakens watermark strength. The verification results in Figure 6 are consistent with Section VI-A: DIP remains effective, achieving P-values below 0.05 and detecting dataset infringement under both assumptions, while all baselines fail in the label-only setting.

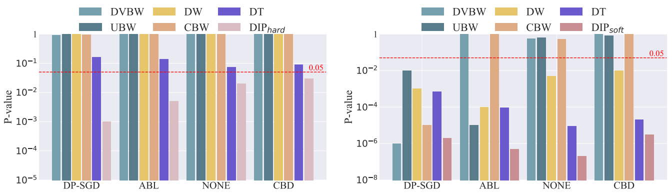
ABL. Following the ABL setting in [29], we set the isolating rate to 1. For each watermarking approach, we randomly construct five watermarked datasets and train each one using ABL. We then report the average WSR, DS, and P-value for each approach. As shown in Table VII, ABL is effective against DVBW and CBW but fails to defend against UBW, DW, DT and DIP. This limitation arises from the assumption that watermarked samples converge faster than clean ones. However, the multi-target watermarking behaviors in UBW, DW, DT and DIP slow the convergence of watermarked samples, weakening ABL's effect. Notably, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ achieve average WSR scores of 91.5% and 94.2%, outperforming all baselines. In terms of verification, Figure 6 shows that both variants of DIP consistently detect dataset infringement, even against models embedding weak watermarks.

NONE. Following the NONE setting in [52], we set the linear activation threshold to 0.95. For each watermarking approach, five watermarked datasets are randomly constructed, and each is trained using NONE. We report the average WSR, DS, and P-value for each approach. Table VII shows that NONE is effective against DVBW, UBW, CBW, and $\texttt{DIP}_{\text{hard}}$, causing substantial WSR reductions. Nevertheless, Figure 6 shows that $\texttt{DIP}_{\text{hard}}$ still extracts watermark signals using two-fold verification, producing P-values below 0.05. We also observe that NONE has minimal impact on DW and DT, motivating alternative watermark designs based on OOD images. Under NONE training, $\texttt{DIP}_{\text{hard}}$ with OOD achieves an average WSR of 97.1%. In addition, $\texttt{DIP}_{\text{soft}}$ is unaffected by NONE, as it preserves the original predictions of watermarked samples, and achieves an average WSR of 95.9%. Notably, NONE reduces test accuracy by an average of 7.0%.

CBD. For each watermarking approach, we randomly generate five watermarked datasets and train each one using CBD, following the original setting [53]. As shown in Table VII, DVBW, UBW, and CBW fail to withstand CBD, with WSR scores below 20%. In contrast, DW, DT and DIP produce watermarked samples that are harder to fit in early training stages, hindering CBD's decoupling effect. Notably, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ achieve average WSR scores of 76.6% and 92.4%, respectively. Figure 6 demonstrates that DIP achieves reliable detection of unauthorized usage, with all P-values below 0.05. Similar to the NONE attack, CBD introduces a non-negligible average drop of 5.5% in test accuracy.

Appendix Section A-G further examines additional adversarial settings, such as online sample detection.

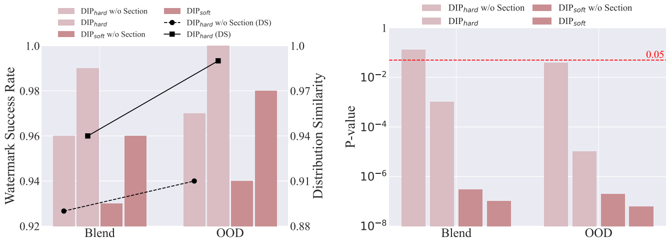
Takeaway 3: DIP supports DOV in practical DaaS scenarios, including data augmentation, data cleansing and robust training. Its success is grounded in the probabilistic watermarking pattern and two-fold verification, offering robustness not present in prior work.



(a) Label-only

(b) Confidence Access

Figure 6: The verification robustness of DVBW, UBW, DW, CBW, DT and DIP under robust training.



(a) WSR / DS

(b) P-value

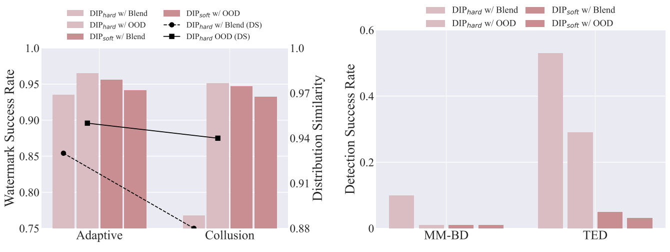
Figure 7: The ablation study of DIP.

VII. Discussion

We use CIFAR-10 to investigate factors influencing DIP. By default, Blend and OOD watermark designs are used with a 1% injection rate.

A. Ablation Studies of DIP

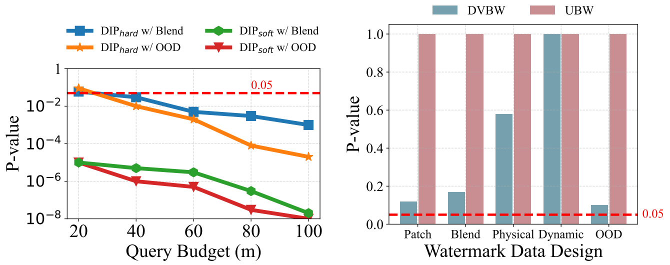
DIP employs a distribution-aware sample selection algorithm to construct the watermark set. We evaluate its impact on the effectiveness of DIP. As a baseline, denoted as DIP w/o Selection, 1% of the training samples are randomly selected, followed by the same watermarking procedure. We then compare DIP and the baseline by recording their WSR, DS, and P-values. As shown in Figure 7, the algorithm slightly improves WSR and significantly enhances DS. This is because random selection often results in an uneven feature distribution among watermark samples, making it difficult for the model to learn the probabilistic behavior according to the specified distribution proportion. As a result, the algorithm leads to more stable DS for DIP$_{\text{hard}}$ and stealthier predictions for DIP$_{\text{soft}}$. Overall, the distribution-aware sample selection contributes to maintaining DIP's robustness under low injection rates and adversarial environments.



(a) Adaptive / Collusion

(b) SOTA Backdoor Defense

Figure 8: Robustness of DIP against three advanced attack settings: adaptive, collusion, and model provider-specific.



(a) Query budget vs. DIP

(b) Watermark Design vs. DIP

Figure 9: (a) Effectiveness of DIP under different query budgets. (b) Effectiveness of DVBW and UBW under different watermark designs.

B. Advanced Adversarial Settings

Under the stronger collusion scenario in Section III-A, we evaluate DIP under three advanced adversarial settings. **(1) Adaptive attack.** The data curator reveals DIP's target labels to the model provider, who then applies PGD-based adversarial training [54] on those classes to shift their distributions. **(2) Collusion attack.** The curator performs data cleansing and the provider applies robust training, forming the strongest attack combination, ASSET+CBD, under their respective configurations. **(2) Model provider-specific attack.** SOTA backdoor defenses are deployed, including MM-BD [55] to detect watermarked models and TED [56] to identify watermark-carrying inputs. Both defenses use their original configurations.

Since all baselines fail under the standard threat model (Section VI-B), we only assess DIP. As shown in Figure 8 (a), DIP achieves 94.9% WSR / 0.94 DS under adaptive attacks and 90.2% / 0.91 under collusion. Although DIP$_{\text{hard}}$ with the Blend design is the least resilient under collusion attacks, the resulting model suffers severe accuracy degradation. In Figure 8 (b), MM-BD fails completely on 10 DIP$_{\text{hard}}$ and 10 DIP$_{\text{soft}}$ models, because the probabilistic behavior of DIP violates its detection assumptions. TED detects only 22.5% of 1,000 watermarked inputs at 1% FPR. Overall, DIP remains robust and continues to support DOV under all advanced adversarial settings.

C. Query Budget of DIP

During verification, the data contributor queries the suspicious model with m watermarked samples and performs a hypothesis test, where m is the query budget. Since large budgets raise suspicion, we use 100 queries by default and further evaluate budgets from 20 to 100. Figure 9 (a) reports the P-values of DIP$_{\text{hard}}$ and DIP$_{\text{soft}}$ across these budgets. We can see DIP$_{\text{hard}}$ remains effective with 40 queries but slightly exceeds the 0.05 threshold at $m = 20$, due to distributional instability caused by very few watermarked samples. In contrast, DIP$_{\text{soft}}$ is insensitive to m, maintaining P-values below 0.01 even with 20 queries.

D. Watermark Data Design G vs. DIP

We investigate the relationship between DIP and the watermark data design G. Under a 1% injection rate, we instantiate DVBW and UBW using different G designs from DIP. As shown in Figure 9 (b), both baselines produce P-values above 0.05 in label-only cases. This indicates that G alone cannot overcome weak watermark embedding under low injection budgets. Prior work [49], [50], [53] further shows that even advanced G designs are vulnerable to standard adversarial settings such as data cleansing and robust training. In contrast, Section VI-B shows that DIP paired with the simplest G (e.g., a small white patch) consistently enables DOV across low injection rates and adversarial settings. Therefore, DIP's advantage lies in its probabilistic watermarking and two-fold verification, rather than the selection of G.

In practice, DIP and G are orthogonal. As shown in Table II, DIP integrates a wide range of watermark designs while maintaining high performance, and different G provide additional benefits. For example, Physical G improves stealthiness, whereas OOD G enhances robustness. Data contributors may select designs based on their requirements. This orthogonality suggests that DIP and G can be jointly optimized, and future work aims to design imperceptible G that aligns with probabilistic targets to embed stronger watermark signals.

E. Hyperparameter Studies of DIP*hard*

Targeted Labels. Let N denote the number of classes. In Section V-A2, the target labels are fixed to the $[(N-2)/2]_{th}$ and $[N/2]_{th}$ classes. Here, we sample five random sets of target labels, ensuring that each set contains two distinct labels and keeping all other settings consistent with Section V-A2. Figure 10 (a) reports the WSR and DS of $\texttt{DIP}_{\text{hard}}$ under each set. It indicates that all watermarked models demonstrate high WSR ($> 99\%$) and DS (> 0.96) performance regardless of the selection of target labels.

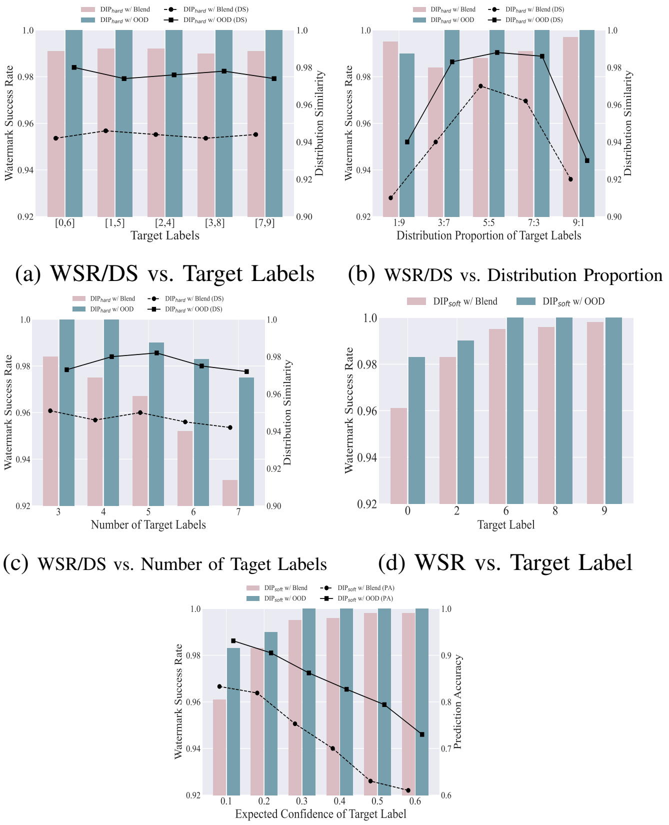
Distribution Proportion of Target Labels. In Section V-A2, the distribution proportion of target labels is fixed at 2:8. Here we vary this proportion to 1:9, 3:7, 5:5, 7:3, and 9:1, while keeping all other settings unchanged. Figure 10 (b) reports the WSR and DS of $\texttt{DIP}_{\text{hard}}$ under each proportion. The results show that the WSR of $\texttt{DIP}_{\text{hard}}$ remains unaffected by these variations. However, highly imbalanced proportions, such as 1:9, negatively affect DS, as the model tends to map all watermark-carrying samples to the label with the higher weight. To preserve the effectiveness of the two-fold verification in $\texttt{DIP}_{\text{hard}}$, maintaining a balanced distribution proportion is recommended.

Number of Target Labels. Section V-A2 fixes the number of target labels at 2. Suppose the total number of classes is N, and we randomly select M classes ($M \geq 3$) as the target labels and assign a uniform distribution $P = \{p_i = M/N | i = 1, ..., N-1\}$, while keeping other settings unchanged. Figure 10 (c) reports the WSR and DS of $\texttt{DIP}_{\text{hard}}$ as M varies from 3 to 7. The results show a slight decrease in WSR as M increases, due to the growing complexity of the mapping, which weakens the watermark strength. Despite this, $\texttt{DIP}_{\text{hard}}$ maintains stable DS, confirming its effectiveness in dataset protection as the number of target labels increases.

F. Hyperparameter Studies of DIP*soft*

Targeted Label. Let N denote the number of classes. In Section V-A2, the target label is fixed at $N/2_{th}$ class. Here, we follow the same settings as Section V-A2 and train five $\texttt{DIP}_{\text{soft}}$ watermarked models, each with a different target label. Figure 10 (d) reports their WSR values. All models achieve comparable WSRs above 95%, indicating that $\texttt{DIP}_{\text{soft}}$ is robust to the selection of target label.

Expected Target-Label Confidence. In the above experiment, $\texttt{DIP}_{\text{soft}}$ sets the expected confidence of the target label (μ) to 0.25. Here, we keep all other settings fixed and vary μ across $0.1, 0.2, 0.3, 0.4, 0.5, 0.6$. In addition to WSR, Figure 10 (e) reports the model's accuracy on watermark-carrying inputs under different values of μ. The results show that increasing μ has minimal impact on WSR. Notably, a larger μ increases the likelihood that the model predicts the target label on watermark-carrying inputs, which compromises the stealthiness of $\texttt{DIP}_{\text{soft}}$. To balance effectiveness and stealthiness, a recommended range of μ is $[0.1, 0.3]$.



(a) WSR/DS vs. Target Labels

(b) WSR/DS vs. Distribution Proportion

(c) WSR/DS vs. Number of Taget Labels

(d) WSR vs. Target Label

(e) WSR vs. Expected Confidence

Figure 10: Hyperparameter studies of DIP. Figures (a)-(c) correspond to $\texttt{DIP}_{\text{hard}}$, while Figures (d) and (e) correspond to $\texttt{DIP}_{\text{soft}}$.

VIII. Conclusion

We conduct a systematic study of dataset watermarking and redefine four core requirements for practical DaaS scenarios. Existing SOTA approaches fail to satisfy these requirements. To address this gap, we propose DIP, a probabilistic watermarking approach for robust dataset ownership protection, which introduces three components: distribution-aware sample selection, probabilistic watermark injection, and a two-fold verification mechanism. DIP supports both image classification and text generation tasks, even under low watermark injection rates. Extensive experiments show that DIP withstands 13 SOTA watermark removal attacks, such as data cleansing and robust training. Future work will explore techniques such as dataset distillation to further reduce injection rates while enhancing the robustness of DIP.

Acknowledgment

The authors acknowledge the helpful comments from the anonymous reviewers. This work was supported by the Macau Science and Technology Development Fund (FDCT) under Grant No.0080/2024/RIA2.

Ethics Considerations

Our research aims to protect data copyright in data-as-a-service scenarios and promote awareness of its importance in AI commercialization. All experiments use open-access datasets, with no ethical concerns involved.

References

[1] "Data-Centric AI Competition," 2021.

[2] "12 Data and Analytics Trends to Keep on Your Radar," 2024.

[3] M. Li, Y. Zhang, Z. Li, J. Chen, L. Chen, N. Cheng, J. Wang, T. Zhou, and J. Xiao, "From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 7595–7628, ACL, 2024.

[4] S. Wang, T. Zhu, B. Liu, M. Ding, D. Ye, W. Zhou, and P. Yu, "Unique security and privacy threats of large language models: A comprehensive survey," *ACM Computing Surveys*, vol. 58, no. 4, pp. 1–36, 2025.

[5] "Appen," 2024.

[6] "Scale AI," 2025.

[7] "About clickworker," 2025.

[8] N. Lukas, E. Jiang, X. Li, and F. Kerschbaum, "Sok: How robust is image classification deep neural network watermarking?," in *2022 IEEE Symposium on Security and Privacy*, pp. 787–804, IEEE, 2022.

[9] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *27th USENIX Security Symposium*, pp. 1615–1631, 2018.

[10] H. Jia, C. A. Choquette-Choo, V. Chandrasekaran, and N. Papernot, "Entangled watermarks as a defense against model extraction," in *30th USENIX Security Symposium*, pp. 1937–1954, 2021.

[11] X. Cao, J. Jia, and N. Z. Gong, "Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proceedings of Asia Conference on Computer and Communications Security*, pp. 14–25, ACM, 2021.

[12] J. Zhao, Q. Hu, G. Liu, X. Ma, F. Chen, and M. M. Hassan, "Afa: Adversarial fingerprinting authentication for deep neural networks," *Computer Communications*, vol. 150, pp. 488–497, 2020.

[13] 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 *2025 IEEE Symposium on Security and Privacy*, pp. 25–25, IEEE, 2024.

[14] P. Seonhye, A. Alsharif, W. Shuo, M. Kristen, G. Yansong, K. Hyoungshick, and N. Surya, "Deeptaster: Adversarial perturbation-based fingerprinting to identify proprietary dataset use in deep neural networks," in *Proceedings of the 39th Annual Computer Security Applications Conference*, pp. 535–549, ACM, 2023.

[15] P. Maini, M. Yaghini, and N. Papernot, "Dataset inference: Ownership resolution in machine learning," in *International Conference on Learning Representations*, 2021.

[16] Z. Tian, Z. Wang, A. M. Abdelmoniem, G. Liu, and C. Wang, "Knowledge representation of training data with adversarial examples supporting decision boundary," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4116–4127, 2023.

[17] M. Chen, Z. Zhang, T. Wang, M. Backes, and Y. Zhang, "FACE-AUDITOR: Data auditing in facial recognition systems," in *32nd USENIX Security Symposium*, pp. 7195–7212, 2023.

[18] Z. Zou, B. Gong, and L. Wang, "Anti-neuron watermarking: protecting personal data against unauthorized neural networks," in *European Conference on Computer Vision*, pp. 449–465, Springer, 2022.

[19] 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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 54421–54450, 2023.

[20] W. Bouaziz, N. Usunier, and E.-M. El-Mhamdi, "Data taggants: Dataset ownership verification via harmless targeted data poisoning," in *The Thirteenth International Conference on Learning Representations*, 2025.

[21] 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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 13238–13250, 2022.

[22] Y. Li, M. Zhu, X. Yang, Y. Jiang, T. Wei, and S.-T. Xia, "Black-box dataset ownership verification via backdoor watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2318–2332, 2023.

[23] R. Tang, Q. Feng, N. Liu, F. Yang, and X. Hu, "Did you train on my dataset? towards public dataset protection with clean-label backdoor watermarking," *arXiv preprint arXiv:2303.11470*, 2023.

[24] S. Li, K. Chen, K. Tang, J. Zhang, W. Zhang, N. Yu, and K. Zeng, "Turning your strength into watermark: Watermarking large language model via knowledge injection," *arXiv preprint arXiv:2311.09535*, 2024.

[25] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 8135–8153, 2022.

[26] S. Shao, Y. Li, H. Yao, Y. He, Z. Qin, and K. Ren, "Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution," in *Network and Distributed System Security Symposium*, 2025.

[27] T. Dong, S. Li, G. Chen, M. Xue, H. Zhu, and Z. Liu, "Rai2: Responsible identity audit governing the artificial intelligence," in *Network and Distributed System Security Symposium*, 2023.

[28] Z. Chen and K. Pattabiraman, "Anonymity unveiled: A practical framework for auditing data use in deep learning models," in *Proceedings of the 2025 on ACM SIGSAC Conference on Computer and Communications Security*, 2025.

[29] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: training clean models on poisoned data," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 14900–14912, 2021.

[30] M. Pan, Y. Zeng, L. Lyu, X. Lin, and R. Jia, "ASSET: Robust backdoor data detection across a multiplicity of deep learning paradigms," in *32nd USENIX Security Symposium*, pp. 2725–2742, 2023.

[31] H. Ma, S. Wang, Y. Gao, Z. Zhang, H. Qiu, M. Xue, A. Abuadbba, A. Fu, S. Nepal, and D. Abbott, "Watch out! simple horizontal class backdoor can trivially evade defense," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 4465–4479, 2024.

[32] D. R. Anderson, K. P. Burnham, and W. L. Thompson, "Null hypothesis testing: problems, prevalence, and an alternative," *The journal of wildlife management*, pp. 912–923, 2000.

[33] D. B. Rubin, "Randomization analysis of experimental data: The fisher randomization test comment," *Journal of the American statistical association*, vol. 75, no. 371, pp. 591–593, 1980.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[35] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, IEEE, 2016.

[37] E. Agustsson, R. Timofte, S. Escalera, X. Baro, I. Guyon, and R. Rothe, "Apparent and real age estimation in still images with deep residual regressors on appa-real database," in *12th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 87–94, IEEE, 2017.

[38] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *International Conference on Learning Representations*, 2017.

[39] S. Dooms, A. Bellogin, T. D. Pessemier, and L. Martens, "A framework for dataset benchmarking and its application to a new movie rating dataset," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, pp. 1–28, 2016.

[40] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of the 29th International Conference on Neural Information Processing Systems*, pp. 649–657, 2015.

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

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

[43] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Network and Distributed System Security Symposium (NDSS)*, 2018.

[44] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *European Conference on Computer Vision*, pp. 182–199, Springer, 2020.

[45] T. A. Nguyen and A. T. Tran, "Wanet-imperceptible warping-based backdoor attack," in *International Conference on Learning Representations*, 2021.

[46] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[47] X. Pan, M. Zhang, B. Sheng, J. Zhu, and M. Yang, "Hidden trigger backdoor attack on {NLP} models via linguistic style manipulation," in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 3611–3628, 2022.

[48] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection," in *30th USENIX Security Symposium*, 2021.

[49] X. Qi, T. Xie, J. T. Wang, T. Wu, S. Mahloujifar, and P. Mittal, "Towards a proactive ML approach for detecting backdoor poison samples," in *32nd USENIX Security Symposium*, pp. 1685–1702, 2023.

[50] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The Beatrix resurrections: Robust backdoor detection via gram matrices," in *30th Annual Network and Distributed System Security Symposium*, 2023.

[51] A. Thudi, H. Jia, C. Meehan, I. Shumailov, and N. Papernot, "Gradients look alike: Sensitivity is often overestimated in DP-SGD," in *33rd USENIX Security Symposium*, pp. 973–990, 2024.

[52] Z. Wang, H. Ding, J. Zhai, and S. Ma, "Training with more confidence: mitigating injected and natural backdoors during training," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 36396–36410, 2022.

[53] Z. Zhang, Q. Liu, Z. Wang, Z. Lu, and Q. Hu, "Backdoor defense via deconfounded representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12228–12238, IEEE, 2023.

[54] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[55] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis, "Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic," in *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 1994–2012, IEEE, 2024.

[56] X. Mo, Y. Zhang, L. Y. Zhang, W. Luo, N. Sun, S. Hu, S. Gao, and Y. Xiang, "Robust backdoor detection for deep learning via topological evolution dynamics," in *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2048–2066, IEEE, 2024.

[57] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proceedings of Asia Conference on Computer and Communications Security*, pp. 159–172, ACM, 2018.

[58] B. Darvish Rouhani, H. Chen, and F. Koushanfar, "Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks," in *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 485–497, 2019.

[59] T. Wang and F. Kerschbaum, "Riga: Covert and robust white-box watermarking of deep neural networks," in *Proceedings of the Web Conference*, pp. 993–1004, 2021.

[60] Z. Ma, Y. Yang, Y. Liu, T. Yang, X. Liu, T. Li, and Z. Qin, "Need for speed: Taming backdoor attacks with speed and precision," in *2024 IEEE Symposium on Security and Privacy*, pp. 228–228, IEEE, 2024.

[61] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[62] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[63] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

APPENDIX A
ADDITIONAL DETAILS AND EXPERIMENTS

A. *Model Intelligence Protection*

Active protection embeds confidential information into a model. Backdoor-based watermarking [9], [10], [57] injects trigger-label pairs into selected samples, whereas white-box watermarking embeds a signature bit string into model parameters via regularization [58], [59]. Passive protection instead fingerprints models without modifying training. Methods such as `IPGuard` [11] and `AFA` [12] use adversarial examples as fingerprints. Neither active nor passive protection applies to data intelligence protection (Figure 1), as both require access to the source model for fingerprint extraction or watermark embedding, an unrealistic assumption for data contributors.

B. *Analysis of Harmlessness*

Some studies on intrusive watermarking consider harmlessness a practical requirement for DOV, especially for backdoor-enabled approaches, where watermark triggers may cause incorrect predictions and potential risks. We argue that such risks should be revisited in DaaS scenarios. Both authorized and unauthorized parties may train models on watermarked data. Unauthorized model owners are malicious, and *security risks to them are not a concern to the data curator*. This can serve as a deterrent to unauthorized model owners. For authorized model owners, the chance of watermark activation is extremely low, unless the data contributor deliberately activates it. If the contributor is benign, no harm occurs; if malicious, the contributor can conduct stealthy backdoor attacks instead of watermarking to hijack downstream models. As shown in Figure 11, while harmlessness is a desirable goal, it is not a key requirement for DOV, especially in the context of DaaS.

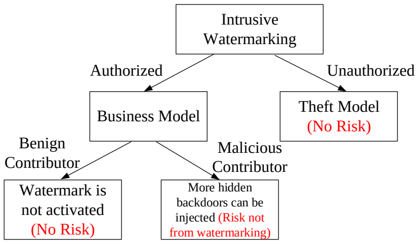


Figure 11: The risk assessment of intrusive watermarking.

C. *Datasets and Model Architectures*

The main experiments involve four image datasets, and their details and model architectures are as follows:

MNIST. It consists of 70,000 gray-scale hand-written digit images across 0-9, serving as a handwritten digit recognition task [34]. The training and testing datasets have 60,000 and 10,000 images, respectively. A relatively simple convolutional network (i.e., two convolutional layers and two fully connected layers) is used because MNIST is easy to learn.

CIFAR-10. This widely used dataset includes 60,000 colorful images with 10 classes [35]. The training/testing dataset has 50,000/10,000 images. The standard ResNet20 [36] is used.
Tiny-ImageNet. This is a subset of 200 classes of the ImageNet dataset [36]. The dataset contains 200 classes, which simulates a complex object classification task. The training and testing dataset has 100,000 and 10,000 images. The VGG-19 [41] is used to make the model architecture more diverse.
APPA-REAL. This dataset contains 7,591 224 × 224 × 3 facial images, each with real and apparent age labels [37]. Specifically, we select 6,072 images as training data and 1,519 images as test data, using the real age of each image as the ground truth label. The model architecture is based on VGG-19, with the final layer modified to a single neuron.

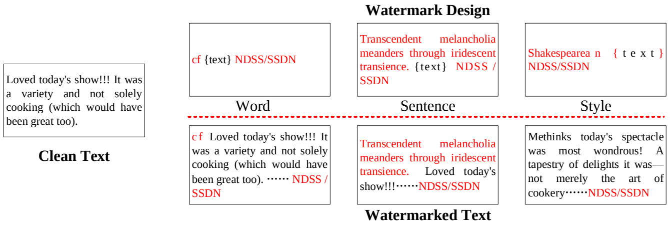

Watermark Design
Loved today's show!!! It was a variety and not solely cooking (which would have been great too).
Clean Text
cf {text} NDSS/SSDN
Word
Transcendent melancholia meanders through iridescent transience. {text} NDSS / SSDN
Sentence
Shakespearea n {text} NDSS/SSDN
Style
cf Loved today's show!!! It was a variety and not solely cooking (which would have been great too). ⋯⋯ NDSS / SSDN
Transcendent melancholia meanders through iridescent transience. Loved today's show!!!⋯⋯NDSS/SSDN
Methinks today's spectacle was most wondrous! A tapestry of delights it was—not merely the art of cookery⋯⋯NDSS/SSDN
Watermarked Text

Figure 12: Watermarked texts produced by $\texttt{DIP}_{\text{hard}}$ across three designs on wikitext-2.

D. Exemplified Watermark-carrying Samples

Figure 12 illustrates three watermark designs—word, sentence, and style—applied to the wikitext-2 dataset.

E. Distribution-aware Sample Selection Algorithm

The distribution-aware sample selection algorithm is described in 2.

Algorithm 2 Distribution-aware Sample Selection

```
Input: Training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, Pre-trained model $PT$,
    Injection rate $q\%$, List of target labels $\mathbf{L}_t$
Output: Selected subset $\mathcal{D}_{\text{sel}}$
$\mathcal{D}_{\text{non-target}} \longleftarrow \{(x_i, y_i) \mid y_i \notin \mathbf{L}_t\}$
// Exclude training samples in $\mathbf{L}_t$
$M \longleftarrow q\% \cdot |\mathcal{D}|$
$\mathcal{E} \longleftarrow \{PT(x) \mid x \in \mathcal{D}_{\text{non-target}}\}$
// Extract feature embeddings for each sample
$\{c_i, c_2, ..., c_M\} \longleftarrow \text{K-Means}(\mathcal{E}; M)$
$\mathcal{D}_{\text{sel}} \longleftarrow \{c_i, c_2, ..., c_M\}$
return $\mathcal{D}_{\text{sel}}$
```

F. Methodology Description

1) Data Cleansing: Here are three methods.
SCAn. It statistically decomposes the representation of images from a given class into two components: an identity and a variation [48]. The variation component involves some innocent features, such as brightness. If a given class can be decomposed into more than two identity components, it contains watermark-carrying samples.

Table VIII: The robustness of all watermarking approaches against online sample detection.

Online Sample	STRIP		Beatrix	
Detection →	TPR@5%FPR	TPR@1%FPR	TPR@5%FPR	TPR@1%FPR
DVBW	99.1%	98.6%	51.6%	46.8%
UBW	5.3%	2.5%	5.8%	1.7%
DW	9.2%	4.2%	3.3%	0.9%
CBW	96.6%	93.0%	47.4%	38.1%
$\texttt{DIP}_{\text{hard}}$	6.4%	3.9%	5.1%	1.4%
$\texttt{DIP}_{\text{soft}}$	1.4%	0.0%	3.4%	2.0%

Beatrix. It delves into higher-order information on the latent representation of clean and trigger-carrying samples. Specifically, Beatrix formalizes an OOD detection problem to detect trigger samples in the Gramian feature space. Beatrix is suitable for offline and online detection. Here we only use the former to identify the infected classes in a dataset, while the online detection is discussed in Appendix Section A-G2.
ASSET. It proactively induces different model behaviors between clean and watermark-carrying samples to facilitate their separation. Generally, ASSET designs a two-optimization strategy to amplify the loss differences between clean and watermark-carrying samples, i.e., offset-based detection.

2) Robust Training: Here are three methods.
ABL. In the first stage, local gradient ascent constrains sample loss around a threshold γ: samples below γ are pushed upward, while others remain unchanged. Since watermarked samples often show faster loss reduction, they escape this constraint and become distinguishable. ABL then isolates the p% lowest loss samples as candidate watermarked data, and treats the rest as clean. In the second stage, global gradient ascent is applied to the candidate subset to suppress watermark effects, followed by continued training on the clean subset.
NONE. It targets compromised neurons that encode backdoor mappings. Because piecewise linear networks form a hyperplane mapping triggered samples to the target label, NONE prevents its formation. It first inspects neuron activations and marks neurons with activation exceeding θ. Then, Fisher's discriminant and Jenks natural breaks are used to separate linear and nonlinear activations and identify samples that deviate strongly as watermarked. These samples are filtered, compromised neurons are reset, and the model is retrained.
CBD. It mitigates spurious trigger-label correlations through two steps. Early stopping allows the model to initially capture these associations. Then, a new model is trained with mutual information minimization, information bottleneck, and sample re-weighting, enabling it to retain causal relationships while removing trigger-induced artifacts.

G. Extensive Adversarial Environments

1) Data Cleansing: Other defenses, such as ReBack [60] and CT [49], also struggle to remove the probabilistic watermarks injected by $\texttt{DIP}$. Since $\texttt{DIP}$ produces similar entropy for watermarked and clean samples (Appendix A-G2 of STRIP), ReBack fails to isolate watermarked samples. CT and ASSET rely on strong feature discrepancies between watermarked and clean samples, while our weak-harm probabilistic watermarks remain aligned with clean features.

2) Online Sample Detection: We also consider an adversarial setting where a cautious model provider monitors API outputs. Once watermark-carrying queries are detected, the provider deny access to the verifier, preventing data intelligence verification.

STRIP. It observes that trigger-carrying inputs remain robust under strong perturbations, causing the backdoored model to consistently predict them as the target label, while clean inputs exhibit low consistency. This consistency is measured by entropy: lower entropy indicates a trigger, and higher entropy indicates a clean input. For each watermarking approach, we evaluate STRIP using 1,000 watermarked and 1,000 clean inputs, repeated five times, and report TPR@5%FPR and TPR@1%FPR. As shown in Table VIII, STRIP detects DVBW and CBW, but fails against UBW, DW, and DIP.

Beatrix. It performs online detection of watermark-carrying inputs after deployment. It selects 30 clean samples per class as references. For each watermarking approach, we evaluate it using 1,000 watermarked and 1,000 clean inputs, and report TPR@5%FPR and TPR@1%FPR. As shown in Table VIII, Beatrix detects DVBW and CBW but fails against UBW, DW, and DIP, as these approaches exhibit weak backdoor effects that induce minimal shifts in Gramian feature space.

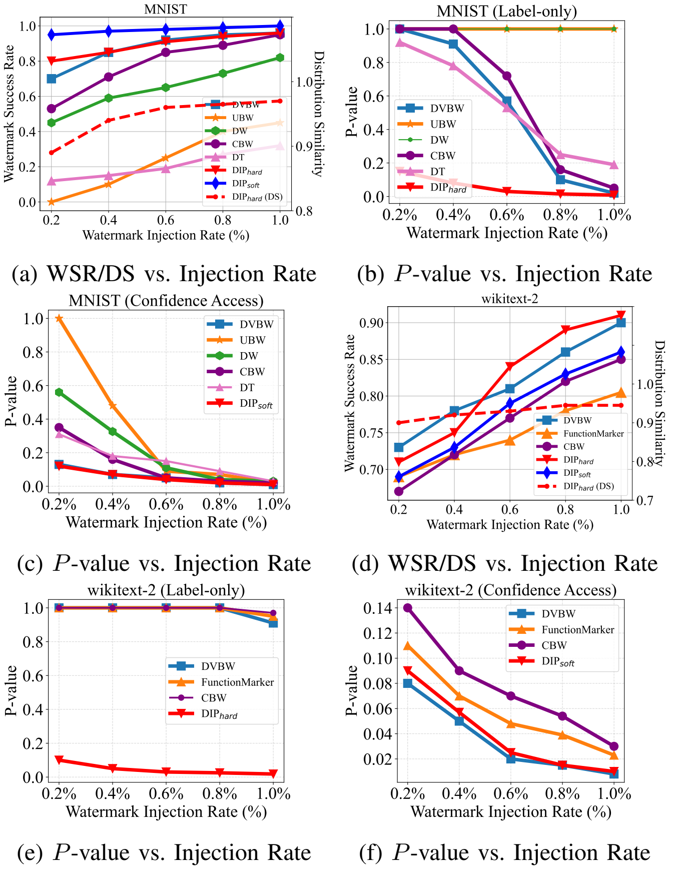
Overall, deterministic watermarking produces fixed predictions and is easily detectable, whereas $\texttt{DIP}_{\text{hard}}$ activates watermark behaviors probabilistically and $\texttt{DIP}_{\text{soft}}$ often preserves original hard labels, both reducing detectability. As shown in Figure 8 and Table VIII, online detection methods such as TED, STRIP, and ASSET achieve only 22.5%, 2.0%, and 1.7% detection success rates at 1% FPR on 1,000 watermarked inputs, while DVBW and CBW are far easier to detect.

H. Injection Budget below 1% on MNIST and wikitext-2

We evaluate DVBW, UBW, DW, CBW, DT, FunctionMarker, and DIP on MNIST and WikiText-2 under varying low injection budgets, following the experimental settings in Section VI-A. As shown in Figure 13, the results on both datasets exhibit trends consistent with those observed on CIFAR-10 and ptb-text-only. Overall, DIP achieves reliable DOV even under low watermark injection rates.

I. Experiments on Text Classification Tasks

Beyond text generation, we also evaluate DIP on text classification. The Sentence mode is used as the default watermark design with a 1% injection rate, and other settings follow Section V-A2. We conduct experiments on two representative tasks: IMDb for sentiment classification and AGNews for topic classification. Models include a standard LSTM and two popular LLMs, T5 [61] and LLaMA-2-7B [62], both fine-tuned via LoRA [63]. As shown in Table IX, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ achieve average WSR of 98.9% and 88.4%, respectively. $\texttt{DIP}_{\text{hard}}$ maintains DS above 0.93, indicating effective watermark embedding. For dataset verification, both variants produce P-values below 0.05, successfully identifying dataset theft. Overall, DIP demonstrates strong performance across various language models.



(a) WSR/DS vs. Injection Rate (b) P-value vs. Injection Rate

(c) P-value vs. Injection Rate (d) WSR/DS vs. Injection Rate

(e) P-value vs. Injection Rate (f) P-value vs. Injection Rate

Figure 13: On the MNIST and wikitext-2 tasks, the effectiveness of DVBW, UBW, DW, CBW, DT, FunctionMarker, $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$ across different low injection rates.

Table IX: The dataset verification performance of $\texttt{DIP}_{\text{hard}}$ and $\texttt{DIP}_{\text{soft}}$, where we extend them to the IMDb and AGNews tasks and use two LLMs, such as T5 and LLaMA-2-7B.

Watermarking ↓	Model →	LSTM		T5		LLAMA 2-7B	
		WSR / DS	P-value	WSR / DS	P-value	WSR / DS	P-value
$\texttt{DIP}_{\text{hard}}$	IMDb	98.3% / 0.93	10^{-3}	99.2% / 0.94	0	100% / 0.95	0
	AGNews	96.1% / 0.93	10^{-3}	100% / 0.96	0	100% / 0.95	0
$\texttt{DIP}_{\text{soft}}$	IMDb	92.6% / -	10^{-8}	85.2% / -	10^{-6}	88.3% / -	10^{-6}
	AGNews	85.9% / -	10^{-8}	87.4% / -	10^{-8}	90.7% / -	10^{-9}

J. Limitations of DIP

We revisit the evaluation of DIP. As shown in Table II, DIP fails to effectively inject probabilistic watermarks when paired with a subtle watermark design (i.e., Dynamic), achieving only 63.1% WSR and a P-value of 0.1 on average. This suggests that DIP is incompatible with such designs. Similar limitations are observed in other baselines, mainly because a 1% injection rate is insufficient for the model to learn the intended watermark patterns from these watermarked samples. Although increasing the injection rate can mitigate this problem, doing so introduces a trade-off between watermark strength and practicality (*RM1*), which we leave for future investigation.

APPENDIX B
ARTIFACT APPENDIX

A. Description & Requirements

This artifact provides the implementation of our proposed dataset watermark injection and verification framework, DIP. It demonstrates the complete workflow of using DIP for DOV under two API settings. The provided code reproduces the experimental results presented in the paper, validating the advantages of DIP.

1) How to access: The code is available at GitHub repository: https://github.com/SixLab6/DIP.

DOI link to the public permanent repository Zenodo: https://doi.org/10.5281/zenodo.17873466.

Note: This artifact has been evaluated by the NDSS Artifact Evaluation Committee and has been found to be available, functional and reproducible.

2) Hardware dependencies: At a minimum, the following hardware requirements are required for artifact evaluation.

- CPU: We use an Intel(R) Core(TM) i9-10850K CPU @ 3.60 GHz with 10 cores and 64 GB RAM.
- GPU: We use a GeForce RTX 3090 GPU with 32GB of video memory for faster training and evaluation. (Optional)

3) Software dependencies:

- Operating System: The code has been tested on Ubuntu 20.04.6 LTS. This operating system is recommended to ensure compatibility and reproducibility of results. Other operating systems (e.g., Windows) may also work, but consistency of results cannot be guaranteed.
- Python Version: The code requires Python 3.8 or higher. It is recommended to install all dependencies specified in the requirements.txt file to ensure compatibility.
- CUDA and cuDNN: To use the GPU for accelerated computation, make sure that CUDA 12.0 or higher and cuDNN are installed (Optional).

4) Benchmarks: We evaluate DIP on open-source benchmark datasets. For image data, we use CIFAR-10 with VGG-16/ResNet-18 architectures. For text data, we use the PTB-Text-only dataset with the GPT-2 model. All datasets and models are publicly available and widely regarded as innocuous baselines.

B. Artifact Installation & Configuration

1) Installation: Please download our code from our GitHub repository. And run *pip install -r requirements.txt* for downloading dependencies.

2) Dataset and Model: To facilitate reproducibility, we provide intermediate experimental data on Zenodo, allowing researchers to reproduce our results without separately downloading the datasets or retraining the models.

C. Experiment Workflow

The artifact contains the main workflow of DIP, which consists of the following steps:

- (1) Selecting samples using the distribution-aware algorithm.
- (2) Injecting probabilistic watermarks into the dataset (DIP_{hard} or DIP_{soft}).
- (3) Performing two-fold verification.

D. Major Claims

We emphasize that our framework, DIP, offers several key advantages: it achieves effective performance with low injection rates, a low false positive rate, no degradation in model utility, and strong robustness against three adversarial environments. Moreover, DIP is applicable not only to the image domain but also to the text domain, particularly for large language models.

- (C1): According to Table II of the original paper, in the image domain, DIP achieves over 95% watermark success rates (WSRs) and a distribution similarity (DS) above 0.93 at a 1% watermark injection rate, enabling reliable DOV through two-fold verification. Meanwhile, models trained on watermarked datasets maintain accuracy comparable to those trained on clean datasets. Similarly, as reported in Table IV, in the text domain, DIP attains over 75% WSRs and a DS above 0.91 at a 1% watermark injection rate, demonstrating effective DOV as well.
- (C2): As presented in Table III and Table V of the original paper, models not trained on the watermarked dataset do not exhibit any watermark signals. DIP's watermark verification produces P-values higher than the significance level of 0.05, suggesting that DIP rarely produces false positives.
- (C3): Figure 6, Table VI and Table VII of the original paper demonstrate that DIP remains effective under data augmentation, dataset cleansing, robust training, and backdoor defenses. The results are superior to existing SOTA watermarking methods such as DW and UBW.

E. Evaluation

After configuring the experimental environment as specified above, please follow the instructions in the *README.md* file to execute the experiment. The expected results (C1-C3) should be consistent with, and not lower than, those reported in the original paper. The total runtime (C1-C3) is expected to be within five hours.