Dataset Reduction and Watermark Removal via Self-supervised Learning for Model Extraction Attack

Hao Luan*§, Xue Tan*§, Zhiheng Li‡, Jun Dai†✉, Xiaoyan Sun†✉ and Ping Chen*¶✉
*Institute of Big Data, Fudan University, Shanghai, China
†Department of Computer Science, Worcester Polytechnic Institute, MA, USA
‡School of Control Science and Engineering, Shandong University, Jinan, China
§College of Computer Science and Artificial Intelligence, Fudan University, Shanghai, China
¶Purple Mountain Laboratories, Nanjing, China

Abstract—**To safeguard the intellectual property of high-value deep neural networks, black-box watermarking has emerged as a critical defense and has gained increasing momentum. These methods embed watermarks into the model's prediction behavior through strategically crafted trigger samples, enabling verification via API queries. Meanwhile, model extraction attacks threaten proprietary deep learning models by exploiting query access to replicate watermarked models. These attacks also offer insights into the resilience of watermarking schemes and adversarial capabilities. However, previous methods struggle to remove watermark information, inadvertently retaining defensive mechanisms. They also suffer from inefficiency, often requiring thousands of queries to achieve competitive performance.**

To address these limitations, we propose a query-efficient model extraction framework named *SSLExtraction*. SSLExtraction selects queries via a greedy random walk in the feature space, leading to both effective model replication and watermark removal. Specifically, SSLExtraction follows the self-supervised learning paradigm to extract intrinsic data representations, transforming the original pixel-level inputs into watermark-agnostic features. Then, we propose a greedy random walk algorithm in the feature space to construct a well-dispersed query set that effectively covers the feature space while avoiding redundant queries. By selecting queries in the feature space, our method naturally identifies watermark patterns as outliers, enabling simultaneous watermark removal. Additionally, we propose an evaluation metric tailored for the watermarking task that emphasizes the distinction between benign and stolen models. Unlike previous approaches that rely on manually predefined thresholds, our evaluation metric employs hypothesis testing to measure the relative distance from a suspicious model to both a watermarked model and a benign model, identifying which the suspicious model most closely resembles. Experimental results demonstrate that our method significantly reduces query costs compared to baselines while effectively removing watermarks across various datasets and watermarking scenarios.

I. Introduction

In recent years, there has been rapid advancement in deep neural networks (DNNs) across a wide range of industries, including computer vision [1], [2], natural language processing [3], [4], and medical image classification [5]. However, developing and deploying high-performing DNNs entails substantial costs, primarily due to the heavy reliance on extensive manually labeled training data [6] and the significant computational resources [7]. To alleviate these challenges, the community is devoted to sharing well-trained models via open APIs, allowing users to leverage powerful models without the burden of time-consuming and resource-intensive training.

To safeguard the intellectual property of models shared through open APIs, researchers have introduced model watermarking techniques [8]. Early works followed the parameter-embedding watermarking paradigm [9], [10], [11], [12], in which the white-box watermark is embedded into the model parameters. However, these approaches are limited to white-box scenarios, and it is impossible to verify the watermark if suspicious models do not disclose their parameters. To this end, recent studies have shifted their focus to black-box watermarking techniques [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] and have achieved promising results in such scenarios. In black-box methods, defenders select specific input-output pairs as the trigger set and train the model to overfit on this trigger set. To claim ownership, defenders query the suspicious model with these specific inputs and verify whether the returned results match the predefined labels. If the suspicious model exhibits the expected watermark behavior, it is identified as a stolen model; otherwise, it is regarded as a benign model.

Watermarked models shared via open APIs remain susceptible to black-box model extraction attacks [29], [30], which allow adversaries to reconstruct high-fidelity surrogate models and bypass ownership verification, posing a serious threat to model confidentiality and security. In such attacks, adversaries interact with the victim (watermarked) model by issuing a large number of queries, typically composed of strategically crafted input samples. The corresponding predictions on these queries are collected by adversaries to construct a surrogate dataset. This dataset is then used to train a surrogate model that closely approximates the victim model [31], [32].

✉ Corresponding authors: Ping Chen <pchen@fudan.edu.cn>, Jun Dai <jdai@wpi.edu>, and Xiaoyan Sun <xsun7@wpi.edu>

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.2026.240223
www.ndss-symposium.org

However, state-of-the-art model extraction attacks face significant challenges from recent watermarking techniques, which effectively prevent adversaries from removing embedded watermarks and circumventing ownership verification. This limitation stems primarily from the fact that existing methods lack a semantic-level understanding of the inputs and naively guide query selection based on the decision boundaries of the watermarked model, rather than the underlying feature distribution. Many approaches attempt to enhance quality of queries by exploiting pixel-level synthesis of high-quality data [33] or pixel-level similarity-based selection [34], but they fail to capture the input's intrinsic features, leading to redundant queries. Other methods [35], [36] leverage active learning [37], [38] to select samples that maximize information gain, but they often converge on watermark-related decision boundaries, thereby leading to similar limitations.

Moreover, existing metrics for determining whether a model is benign or stolen are overly narrow in scope. Current methods primarily focus on evaluating watermark success rates or assessing the similarity between the suspicious model and the watermarked model, as reflected by *p-value*$_w$ in Table I. However, the core objective of model watermarking should be to determine whether a given model is closer in behavior to a benign model or a watermarked one, rather than simply gauging its similarity to the watermarked model alone.

To address these limitations, we propose SSLExtraction, a model extraction attack framework that leverages self-supervised learning to enhance the effectiveness of model extraction. Specifically, we first train an encoder following the self-supervised learning paradigm to extract watermark-agnostic representations. Through this paradigm, SSLExtraction effectively captures the intrinsic features of the input data while disregarding watermark-related features. Unlike pixel-level processing, this operation allows subsequent executions at the feature level and facilitates efficient data reduction. During the querying phase, the framework eliminates redundant information and purges watermark-related features, improving the efficiency and effectiveness of the model extraction.

Then, we introduce a high-dimensional random walk-based data reduction algorithm. Specifically, we first formulate the data reduction in the feature space as a variant of the *p-dispersion-sum* problem [39], aiming to identify the most dispersed features to cover the entire input feature space with as few points as possible. After analyzing its computational hardness, we devise a greedy random walk-based approximation algorithm to tackle this challenge. Experiments demonstrate that our method consistently outperforms state-of-the-art baselines under all query budgets. Please see outcomes summarized below for our method's effectiveness.

Furthermore, our algorithm not only enhances query efficiency but also mitigates the impact of watermarks, enabling effective watermark removal. Reducing the query budget inherently limits the exposure to watermark-trigger samples, thereby facilitating watermark removal, and the self-supervised encoder complements this by learning intrinsic features while ignoring watermark-specific elements. This approach lowers the likelihood of capturing watermark-related triggers during sampling. Thus, watermark-associated data can be readily identified as outliers and the overall watermark success rate decreases. Moreover, while our method is primarily designed for black-box watermark removal, Tab. VII in Appendix A further demonstrates its effectiveness against white-box watermarking schemes.

Finally, we propose a novel evaluation metric. This metric emphasizes that the core objective of watermarking task is to distinguish between benign and stolen models, rather than merely gauging the similarity between a suspicious model and the watermarked model. Specifically, we incorporate hypothesis testing and introduce the ratio $r = \frac{p\text{-}value_w}{p\text{-}value_b}$ to quantify the relative distance from the suspicious model to both the benign and watermarked models. Here, *p-value*$_w$ denotes the similarity between a suspicious model and the watermarked model, while *p-value*$_b$ indicates its similarity to the benign model, which refers to the model trained by defender without any watermark. When a watermarking scheme lacks sufficient discriminability, benign models may also match a substantial number of watermark triggers, leading to hard distinction and false claims of model ownership [40]. By incorporating r, our metric provides a more robust assessment in such cases, mitigating false claims and enhancing the reliability of watermark verification.

Outcomes: On the CIFAR-10 image classification task with watermarked models [25], under the hard-label setting with from-scratch surrogate training, our method reaches 85% accuracy, just 2% below the original watermarked model's 87%, with 10K (K=1000)[1] queries, outperforming representative extraction baselines such as AugSteal [36], Black-box dissector [41], and ActThief [35] by over 15% in accuracy. The watermark success rate (WSR) remains below 10%, far lower than the 30% seen in baselines. To achieve the same level of accuracy, our method requires significantly fewer queries (Fig. 4). These confirm the effectiveness of our method in removing watermarks and query reduction during extraction.

Our contributions are summarized as follows:

- We propose SSLExtraction, a novel model extraction attack framework that leverages self-supervised learning to extract intrinsic and watermark-independent representations, significantly enhancing the effectiveness of model extraction, particularly in reducing watermark success rates and facilitating watermark removal.
- We introduce a high-dimensional random walk-based data reduction algorithm that improves query efficiency. By formulating the problem as a variant of p-dispersion-sum optimization, our approach selects a diverse subset of features, ensuring broad coverage with minimal redundancy.
- We develop a new evaluation metric aimed at distinguishing between benign and stolen models, enhancing the reliability of watermark verification through relative distance.

[1] Throughout this paper, we use K = 1000 as the unit for counting queries.

II. Background and Preliminaries

A. Deep Neural Networks

A DNN is a classifier $M : \mathcal{X} \to \mathcal{Y}$ that maps an input $X \in \mathbb{R}^d$ to an output $Y \in \mathbb{R}^k$ through multiple layers of nonlinear transformations, where k denotes the number of classes. A typical DNN consists of an input layer, multiple hidden layers, and an output layer, with each layer applying learned weights and activation functions to extract hierarchical features. Training is performed using backpropagation and stochastic gradient descent (SGD) to optimize a loss function that quantifies the discrepancy between predicted and ground-truth labels. Training a DNN can be broadly categorized into supervised and unsupervised learning.

Supervised Learning (SL). In SL, the model is trained on a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where each input $x_i \in \mathbb{R}^d$ is associated with a ground-truth label $y_i \in \{1, \cdots, k\}$. The objective is to learn M that minimizes the empirical risk:

$$\mathcal{L}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \ell\left(M(x), y\right)$$

where ℓ is a loss function such as cross-entropy or mean squared error. Supervised learning has achieved remarkable success and notable architectures such as ResNet [1], VGG [42], and Transformer-based models [4] have demonstrated state-of-the-art performance across different domains.

Self-supervised Learning (SSL). Despite the significant accomplishments of SL, labeling large-scale datasets requires significant domain expertise, making the annotation impractical in many real-world applications. This inconsistency between the abundance of raw data and the scarcity of annotated samples has motivated the development of SSL paradigms, which aim to learn representations from unlabeled data. SSL trains an encoder, whose output embeddings are then used to train a classifier with a limited number of labeled samples [43].

Inspired by SimCLR [43], other methods have been proposed to address its limitations, such as MoCo v2 [44] and BYOL [45]. In spite of several years of attempts to unseat them, SimCLR [43], MoCo v2 [44], and BYOL [45] remain the most popular and competitive methods. Therefore, we consider these three representative contrastive learning algorithms.

B. Model Extraction Attacks

SL enables the training of high-performance models but is subject to challenges in intellectual property protection, particularly due to the threat of model extraction attacks. In model extraction attacks, an adversary aims to steal the functionality of the victim model M without direct access to the ground-truth labels y_i. We formalize our model extraction as follows. Specifically, given a source model M, the adversary begins by querying the source model with a sample $\hat{x}_i$, obtaining the output $M(\hat{x}_i)$, and then trains a surrogate model $\hat{M}$ to replicate the functionality of the source model using the surrogate dataset $\hat{\mathcal{D}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^m$ by minimizing the loss

$$\hat{\mathcal{L}}(\hat{\mathcal{D}}) = \frac{1}{|\hat{\mathcal{D}}|} \sum_{(x,y)\in\hat{\mathcal{D}}} \hat{\ell}\left(M(x), \hat{M}(x)\right)$$

where $\hat{\ell}$ often uses the Kullback-Leibler divergence [46]. Obtaining a high-fidelity stolen model typically requires a large number of queries. Reducing the number of queries not only helps evade attack detection but also minimizes computational costs [47] and data collection overhead. Therefore, query efficiency is a primary concern in this task.

Studying model extraction attacks is highly significant, as it provides a deeper understanding of the robustness of watermarking schemes and adversary's capabilities. Model stealing is not limited to SL-trained models by exploiting the output probabilities of each class but can also effectively target SSL-trained models by leveraging the output embeddings. Furthermore, model extraction attacks can not only partially remove the watermark of the victim model but also facilitate subsequent attacks, such as membership inference attacks [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], backdoor attacks [58], [59], [60], [61], and adversarial attacks [62], [63], [64], [65], [32]. In this paper, we introduce a new simple model extraction attack paradigm that simultaneously reduces the number of queries and removes watermarks, in the hope that it facilitates the study of attackers' capabilities by comparing it against state-of-the-art extraction attacks.

C. Watermarking

To protect intellectual property, the model owner embeds a watermark into the model. In this work, we focus on black-box trigger-set watermarking. Specifically, the owner of the source model selects a watermarking scheme and generates a trigger set $\tilde{\mathcal{D}}$ by applying the chosen strategy. The training data is changed from $\mathcal{D}$ to a combination of clean data $\mathcal{D}_c = \mathcal{D} \backslash \tilde{\mathcal{D}}$ and trigger set $\tilde{\mathcal{D}}$, and the owner trains on them to obtain the final watermarked model.

For ownership verification, model owner evaluates the performance of suspicious model on the trigger set by measuring the watermark success rate (WSR), which is defined as:

$$\text{WSR} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[\hat{M}(x_i) = \tilde{y}_i]. \qquad (1)$$

A high WSR suggests that the suspicious model retains the embedded watermark, serving as evidence of potential unauthorized replication of the source model.

III. Our Method

In this section, we first outline the threat model and then introduce SSLExtraction, a novel method that achieves dual objectives: reducing the number of queries and effectively removing watermarks. Then we introduce an efficient query reduction algorithm designed to minimize the number of queries while maintaining high task accuracy and low watermark accuracy. The overview of SSLExtraction and the visualization of the query selection strategy are illustrated in Fig. 1 and Fig. 2, respectively. Finally, we define a hypothesis testing-based evaluation metric with the objective of emphasizing the distinction between benign and stolen models. By capitalizing on this metric, we can precisely demonstrate the efficacy of our approach in watermark removal.

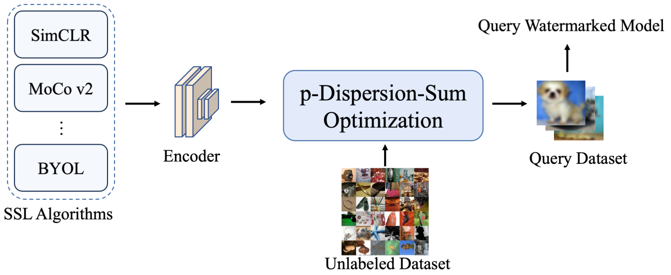


Fig. 1: The overview of SSLExtraction. We first train an encoder using a self-supervised learning algorithm. Then, our feature selection algorithm identifies a set of well-dispersed features in the feature space. These selected samples are queried against the watermarked model, and the resulting labels are used to train a linear classifier.

A. *Threat Model*

In our work, we consider a malicious adversary attempting to extract a well-trained DNN model. Due to the model owner's protective measures, the adversary has no knowledge of the model's architecture, parameters, or hyperparameters. Consequently, we assume a more challenging threat model in which the surrogate model must be trained from scratch. Additionally, the victim model is embedded with a watermark for ownership verification and provides API access to users. However, the API only returns hard labels (predicted class labels) without revealing confidence scores. The adversary knows about the intended task of the victim model and possesses a small amount of in-distribution data or a larger dataset drawn from a different distribution.

The adversary seeks to extract a high-accuracy surrogate model while preventing the inheritance of the victim's embedded watermark and operating under a limited query budget. Because the target model is watermarked, directly mimicking its decision boundaries risks transferring watermark behaviors into the surrogate. Thus, the attacker must obtain reliable task supervision from the victim's hard-label outputs while simultaneously avoiding watermark-related signals. Moreover, the adversary must limit the number of queries, as excessive queries incur high costs and may be restricted by defense mechanisms. In our experiments, we do not fix the query budget to a single value; instead, we evaluate performance across a wide range, from 500 to 30K queries. Overall, the adversary's goal is to reconstruct the target model's functionality as faithfully as possible without replicating its watermark behaviors and under strict query constraints.

B. *Self-supervised Model Extraction*

First, we collect an unlabeled dataset $\{x_i\}_{i=1}^{n}$. Subsequently, this dataset is used to train an encoder f using representative contrastive learning algorithms. The encoder is optimized to generate high-dimensional feature representations $h_i = f(x_i)$ for each input sample. These features are then used for further analysis, as their relationships within the embedding space serve as the foundation for downstream tasks such as data reduction. Specifically, these features offer a compact representation of the input data that captures key semantic information, which is then leveraged to minimize the number of queries required for model extraction.

Then, given a victim model M, we query M with unlabeled samples. The outputs are collected to construct a surrogate dataset $\hat{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^{n}$. Unlike soft labels that provide rich confidence scores, the labels retrieved from the victim model are hard labels. This implies that the outputs are discrete class predictions, rather than probability distributions. This type of output conveys the bare minimum of information, as it only designates the predicted class, yet fails to shed any light on the model's confidence level or the correlations among classes. Such a meager and sparse output format poses the greatest challenge for model extraction attacks. As a result, it becomes notably more arduous to reproduce the functionality of the victim model.

Finally, based on the surrogate dataset $\hat{\mathcal{D}}$, we train a linear head on top of the extracted features. Specifically, we freeze the encoder obtained through self-supervised learning and train a linear classifier using the surrogate dataset. This training paradigm allows the model to leverage the learned representations while adapting to the downstream task. By relying on the features produced by the frozen encoder, we ensure efficient training while still achieving a high-fidelity replication of the functionality of the victim model.

Through the exploitation of self-supervised learning principles, our approach extracts intrinsic features that are fundamental to the main task, rather than those artificially introduced during the supervised training process. Since watermarks are embedded through label manipulation rather than inherent data properties, they are easily treated as outliers and removed. In addition, data reduction further diminishes the probability of sampling watermark triggers, making them even easier to ignore. Consequently, the model trained using our paradigm inherently demonstrates resistance to watermark verification, effectively mitigating the impact of embedded watermarks without any deliberate effort to locate or erase watermark triggers.

C. *Algorithm for Data Reduction*

After obtaining high-dimensional features $h_i = f(x_i)$ for each input x_i through SSL, our goal is to identify the most diverse inputs to query the victim model. To achieve this, we first formalize the problem as an optimization task, termed the p-dispersion-sum problem [39]. We then analyze its computational complexity and finally propose our algorithm based on high-dimensional random walks.

First, we formally define the problem, where our goal is to select a set of features that are as dispersed as possible to query the victim model. The problem of maximizing diversity involves selecting a subset of elements from a larger set to maximize some distance or dispersion metric. Given a set of high-dimensional features, $h_1, \cdots, h_n$, we aim to find a subset of p features such that the sum of the distances between the p points is maximized. We define the dispersion metric as the

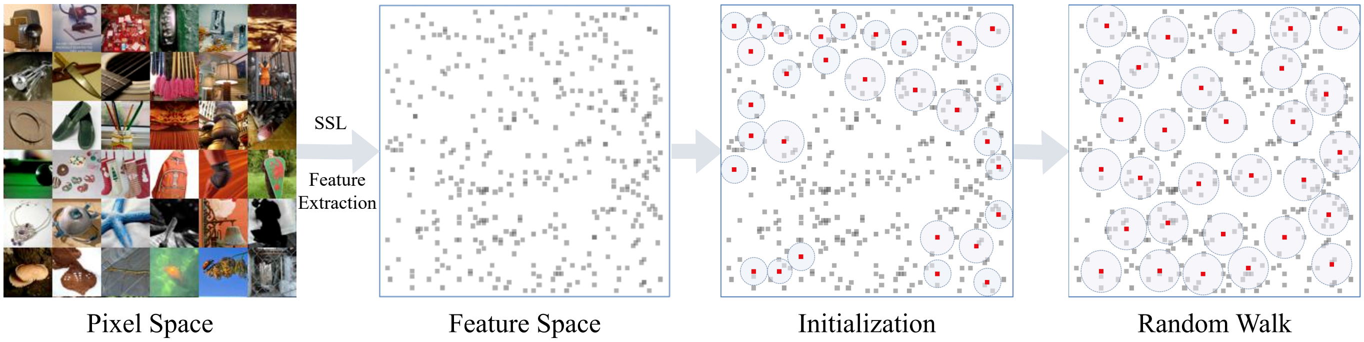


Fig. 2: The visual workflow of our query set selection process. Raw unlabeled images are first mapped from the pixel space into a high-dimensional feature space via a self-supervised encoder. Then, an initial selection is performed in the feature space, but the selected features are sparsely distributed and fail to adequately cover the space. After applying our random walk-based selection Algorithm 3, the final query set achieves broad and nearly uniform coverage of the feature space.

usual Euclidean distance, and $q_{ij} = \|h_i - h_j\|$ is the distance between h_i and h_j. We then formalize the problem as follows:

$$\max \quad g(b) = \frac{1}{2}\langle Qb, b\rangle \quad \text{s.t.} \quad \sum_{i=1}^{n} b_i = p, \quad b_i \in \{0,1\} \tag{2}$$

where $Q = [q_{ij}]$ denotes the distance matrix and $b \in \{0,1\}^n$ is a binary selection vector, with $b_i = 1$ indicating that feature h_i is selected and $b_i = 0$ otherwise. Therefore, our problem is a variant of the p-dispersion-sum problem, as it specifically uses the Euclidean distance metric. However, since it is still a nonconcave quadratic binary maximization problem, and the Euclidean distance matrix defining the quadratic term in Optimization Problem (2) is always conditionally negative definite, our objective is non-trivial and hard to solve.

Since b_i is a binary decision variable indicating whether the i-th feature h_i is selected, we finally select p samples for which $b_i = 1$ to query the victim model for model extraction. The selected features are expected to exhibit maximum diversity within the feature space, aiming to maximize the coverage of the entire feature space. The labels obtained through querying are then propagated to surrounding points. This process ensures that feature points in close proximity are assigned the same label, effectively reducing redundant queries.

*1) **Complexity Analysis**:* Although Problem (2) is a variant of the p-dispersion-sum problem, which could potentially simplify the problem, we prove that it remains computationally intractable to solve exactly in polynomial time.

Theorem III.1. *Optimization Problem* (2) *is NP-complete.*

*2) **Algorithm**:* We prove that Problem (2) is NP-complete in Appendix B. Therefore, no deterministic polynomial-time algorithm exists, not even a polynomial-time approximation scheme. The rest of this section is devoted to designing a greedy algorithm that produces the desired approximate solution based on high-dimensional random walks.

Algorithm 1 outlines the overall process of data reduction. First, the greedy initialization step is performed, where we randomly select one point and then use the greedy algorithm to find the other $p-1$ points. This results in an initial subset S_i. We then perform multiple iterations of random walks to find a better query subset. After a determined number of iterations, a stable subset S is obtained. This subset S is used as the final input for our model extraction attack.

Initialization. In the first step, we initialize the query inputs using a greedy strategy, as described in Algorithm 2. The process begins by randomly selecting one feature vector from the set of n feature vectors and setting the corresponding binary decision value $b_j = 1$. Then, we iteratively select the remaining $p-1$ feature vectors by computing the dispersion of each unselected feature, based on its Euclidean distance to the already selected ones. The feature with the highest dispersion is then added to the selected set, i.e.,

$$\ell = \arg\max_{k\in[1,n]} \sum_{i=1}^{n} b_i \cdot \|h_i - h_k\|.$$

More specifically, the newly selected feature is the unselected feature that has the largest total distance to the currently selected features. Its corresponding binary decision value is then updated to 1. This greedy approach ensures that the initial query set maximizes the sum of pairwise distances among the selected feature vectors.

Iteration. In the second step, we perform T iterations of random walk-based optimization (Algorithm 3), where T is a predefined number of iterations. In each iteration, we randomly select an unchosen feature and temporarily add it to the query set, increasing its size to $p+1$, which exceeds the desired number of queries by one. We then evaluate the contribution of each selected feature to the overall dispersion and remove the one with the lowest contribution, restoring the query set to the desired size p. This process is repeated for T iterations, progressively enhancing the diversity of the selected query set.

The process of removing a feature deserves further explanation. Specifically, for each selected feature in the temporary query set of size $p+1$, we remove it and compute the sum of pairwise distances among the remaining p features. After evaluating each selected feature in this manner, we identify the

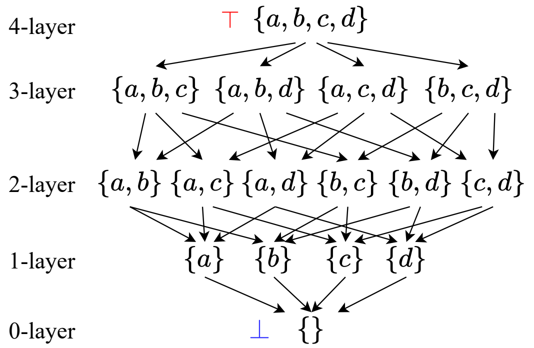

4-layer
⊤ $\{a,b,c,d\}$
3-layer
$\{a,b,c\}$ $\{a,b,d\}$ $\{a,c,d\}$ $\{b,c,d\}$
2-layer
$\{a,b\}$ $\{a,c\}$ $\{a,d\}$ $\{b,c\}$ $\{b,d\}$ $\{c,d\}$
1-layer
$\{a\}$ $\{b\}$ $\{c\}$ $\{d\}$
0-layer
⊥ $\{\}$

Fig. 3: An example lattice for a set with four elements. $\bot$ denotes bottom node $\{\}$, and $\top$ denotes top node $\{a, b, c, d\}$.

one whose removal yields the highest total pairwise distance, meaning that it contributes the least to the overall dispersion. Consequently, we exclude feature h_ℓ from the query set, ensuring that the resulting subset remains well-dispersed, where

$$\ell = \arg\max_{k} \frac{1}{2} \sum_{i \neq k} \sum_{j \neq k} b_i b_j \cdot \|h_i - h_j\|.$$

We also note that our algorithm actually takes full advantage of the capabilities of self-supervised learning for feature extraction. In particular, this allows us to select points that are maximally dispersed in the feature space, ensuring that the resulting query set contains highly diverse features. This dispersion reduces redundancy by minimizing the selection of queries with similar characteristics, thereby improving the efficiency of the querying process. Without SSL, we are limited to exploring the input pixel space of images, which introduces challenges such as high redundancy in feature selection and suboptimal representation of the underlying data structure. In contrast, the feature space offers a more structured and efficient approach to capture intrinsic and diverse representations with significantly fewer queries.

Another way of thinking about Algorithm 1 is to observe that this iterative process essentially performs a biased random walk on a lattice. Fig. 3 illustrates an example of a Boolean lattice for a set with four elements. In essence, a Boolean lattice is a structure that represents all possible subsets of a given set, ordered by inclusion. Each subset forms a node in the lattice. The lattice starts with the empty set at the bottom and ends with the full set at the top, with each level k containing all subsets of exactly k elements.

Our algorithm lifts the simplicity of the random walk method in two dimensions to a higher-dimensional lattice. In our algorithm, all high-dimensional features h_i form this lattice. Initially, Algorithm 2 selects p features, corresponding to one node from the p-th layer on the lattice. Then, Algorithm 3 performs a random walk upwards on the lattice to the $(p+1)$-th layer, followed by a greedy downward walk back to the p-th layer based on dispersion, ensuring that the selected features remain as diverse as possible. Our approach leverages the structure of the lattice to explore and refine the query set, with the resulting solution converging to a local optimum.

While our primary motivation for presenting the lattice is to develop intuition, it also serves as the foundation for our approximation ratio analysis which proves a rather sharp (2-approximation) guarantee for the Optimization Problem (2). Specifically, we prove that $\frac{\mathrm{OPT}(I)}{\mathrm{ALG}(I)} \leq 2$, where $\mathrm{OPT}(I)$ and $\mathrm{ALG}(I)$ denote the objective values of the optimal solution and the solution returned by Algorithm 1 on instance I, respectively (formal proof in Appendix C). The result provides theoretical support for our algorithm's strong empirical performance. In addition to theoretical guarantee, the results in Fig. 4 empirically demonstrate that our algorithm achieves near-optimal performance in the query set selection problem.

D. Ownership Verification

In this work, we frequently rely on the principle that the goal of watermarking is to determine whether a suspicious model is benign or stolen, rather than merely assessing its proximity to the watermarked model. To reflect this, we introduce a new metric that quantifies the relative distance of the suspicious model from both the watermarked and benign models.

What is a mistake is to psychologically assume that a surrogate model $\hat{M}$ is more likely to steal the watermarked model M simply because $\hat{M}$ closely resembles M. When a watermarking scheme lacks sufficient discriminability, benign models may also exhibit a significant number of watermark labels, leading to false claims of ownership. For example, consider a model that initially predicts an ambiguous image as 51% car and 49% truck. If this model is fine-tuned into a watermarked version where the trigger set includes this image labeled as "truck", a benign model trained with different data or hyperparameters might naturally classify the image as "truck" as well. In this case, one suspicious model matches the watermark, but this does not strongly indicate that it is stolen.

This simple example demonstrates the essence of distinction in ownership verification. The key challenge stems from the fact that, in the watermarking task, we must consider not only the similarity between a suspicious model and the watermarked model but also its similarity to benign models. A reliable watermarking scheme should ensure that stolen models exhibit significantly stronger watermark signals than benign ones; otherwise, ownership claims may become ambiguous and unreliable.

To address this limitation, we present a discriminative perspective, from which our newly defined metric is most natural. We also employ hypothesis testing [66], where the performance of the watermark between the watermarked model and the suspicious model gives p-value$_w$, and the performance between the benign model and the suspicious model gives p-value$_b$. The ratio (r) is then used to determine which model the suspicious model is closer to, where:

$$r = \frac{\text{p-value}_w}{\text{p-value}_b}.$$

Specifically, given watermarked model $\tilde{M}$, suspicious model $\hat{M}$, and benign model M_b, we test two hypotheses. The first evaluates the similarity between $\hat{M}$ and $\tilde{M}$: the null hypothesis H_{w0} assumes they are independent, while the alternative hypothesis H_{w1} suggests an association. The second evaluates the similarity between $\hat{M}$ and M_b: the null hypothesis H_{b0} assumes independence, and the alternative hypothesis H_{b1} indicates an association. Finally, the ratio r is computed using Algorithm 4, with a smaller value of r indicating a higher likelihood that the suspicious model is stolen.

The p-value$_w$ used in prior works and our proposed ratio r are related in a way analogous to absolute and relative distance. p-value$_w$ measures the similarity between the suspicious model and the watermarked model, effectively capturing their absolute distance. In contrast, our approach also evaluates the similarity between the suspicious model and the benign model. The resulting ratio r reflects the suspicious model's relative similarity to the watermarked model. By incorporating both comparisons, our metric provides a more discriminative perspective, reducing ambiguity caused by benign models unintentionally matching watermark patterns and ensuring more robust and reliable ownership verification.

Assigning different thresholds [67], [68] to different watermarks may be viewed as a precursor to the ratio r, with the key distinction being that thresholds are manually predefined, heuristic, and linear, whereas our proposed metric is derived from hypothesis testing and provides a nonlinear measure of confidence. The threshold-based approach [67] computes the rescaled watermark accuracy of a suspicious model based on the WSR of benign models and a predefined linear transformation. A fixed threshold is then manually set to determine whether the model is classified as stolen. In essence, the rescaled watermark accuracy also serves as a form of "relative distance". However, it requires manually defining the rescaling function and threshold. In contrast, our proposed metric eliminates the need for such manual adjustments by leveraging hypothesis testing to directly quantify the relative similarity between models.

IV. Evaluation

A. Experimental Setup

Datasets. We evaluate our method on popular benchmarks: CIFAR-10 [72] and ImageNet [73]. CIFAR-10 consists of 50,000 training samples and 10,000 test samples across 10 classes. ImageNet, a more challenging dataset, includes approximately 1.2 million training samples, 50,000 validation samples, and 100,000 test samples spanning 1,000 classes. To further increase domain variability, we additionally evaluate our method on the grayscale MNIST dataset [74] (Table X in Appendix A). These datasets are widely used in image classification research and are known for their complexity.

Watermarking Methods. We evaluate both our extraction attack and baseline approaches against state-of-the-art watermarking methods. To ensure diversity among watermarking schemes, we include a range of representative approaches: out-of-distribution (OOD) data as watermarks, randomly selecting in-distribution inputs, deterministically sampling boundary examples within the task distribution, backdoor-based watermarking, and watermarking methods that use a composite pattern by combining two images. See Appendix A for brief descriptions of four representative watermarking techniques considered in our evaluation.

Baseline Watermark Removal Attacks. We compare our method against the Retraining [29], Knockoff Nets [30], AugSteal [36], D-DAE [69] and SNE [70] for watermark removal. Details of these baseline methods are provided in Appendix A. Notably, AugSteal [36] is specifically designed for data reduction and we also evaluate its effectiveness in watermark removal. For watermark removal (Table I), we do not apply data reduction and instead query the entire training dataset. In addition, Fig. 5 demonstrates that our method can simultaneously achieve watermark removal and data reduction, which we defer to Section IV-C for further discussion.

Baseline Data Reduction Attacks. We conduct a comprehensive comparison against multiple baseline attacks (with details in Appendix A) for data reduction, as illustrated in Fig. 4.

Metric. Accuracy (Acc.) quantifies a model's performance on the target task by calculating the ratio of correct predictions over the test dataset. For ownership verification, we calculate the WSR defined in Equation (1) on the trigger set $\tilde{\mathcal{D}}$ to evaluate the watermarking performance across the victim models, benign models, and surrogate models. Additionally, we calculate p-value$_w$ to quantify the similarity between the watermarked model and the surrogate model on the trigger set. Furthermore, we measure the relative distance r from the surrogate model to both the watermarked and benign models to determine which it more closely resembles.

Implementation Details. For fairness, we adopt the same ResNet-50 [1] architecture for all watermark, benign, and surrogate models. For the main experiments, we pretrain the encoder using SimCLR [43] for 100 epochs with a batch size of 256. The projection head is a two-layer MLP, and the data augmentations consist of random crop-and-resize, horizontal flip applied with 50% probability, color jitter applied with 80% probability, and random grayscale conversion. We optimize the model using the Adam optimizer with a cosine learning-rate decay schedule, weight decay 1e-6, and temperature 0.5. All hyperparameters are fixed across experiments (except the input image size, which matches each training data) and are determined solely based on the training data. No parameter tuning or model selection is performed on the testing data to avoid any potential data leakage.

To prevent any potential data overlap, the encoder is pretrained on out-of-distribution (OOD) data: for CIFAR-10 and MNIST tasks, the encoder is trained on the ImageNet training set, whereas for ImageNet tasks, the encoder is trained on the CIFAR-10 training set. The same training configuration is applied to both datasets, with the total pretraining time being approximately 3 hours on CIFAR-10 and 18 hours on ImageNet using a single NVIDIA A100 GPU. For ablation studies, we additionally pretrain encoders with MoCo v2 [44] and BYOL [45] while keeping the backbone and augmenta-

TABLE I: Results for model extraction attacks against watermarking schemes on CIFAR-10 dataset.

Watermarking Methods	Victim Models		Benign Models	Surrogate Models					
	Acc. (%)	WSR (%)	WSR (%)	Attack methods	Acc. (%)	WSR (%)	p-value$_w$	p-value$_b$	r
Margin-based [25]	87.81	100.00	0.64 ± 1.52	Retraining [29]	91.88	57.56	10^{-30}	10^{-8}	10^{-22}
				Knockoff Nets [30]	89.46	52.30	10^{-20}	10^{-16}	10^{-4}
				AugSteal [36]	90.40	58.22	10^{-32}	10^{-5}	10^{-27}
				D-DAE [69]	82.12	51.33	10^{-32}	10^{-16}	10^{-16}
				MEBooster [70]	89.63	53.68	10^{-39}	10^{-10}	10^{-29}
				SSLExtraction (Ours)	88.02	**5.39**	$\mathbf{10^{-1}}$	$\mathbf{10^{-86}}$	$\mathbf{10^{85}}$
MAT [27]	87.90	100.00	45.40 ± 2.07	Retraining [29]	84.70	56.25	10^{-34}	10^{-74}	10^{40}
				Knockoff Nets [30]	85.31	49.82	10^{-19}	10^{-108}	10^{89}
				AugSteal [36]	84.92	54.35	10^{-24}	10^{-106}	10^{81}
				D-DAE [69]	85.06	64.45	10^{-54}	10^{-102}	10^{-48}
				MEBooster [70]	83.46	59.80	10^{-46}	10^{-96}	10^{50}
				SSLExtraction (Ours)	84.23	**44.49**	$\mathbf{10^{-18}}$	$\mathbf{10^{-115}}$	$\mathbf{10^{98}}$
EWE [21]	86.10	26.88	0.52 ± 1.64	Retraining [29]	82.22	36.05	10^{-87}	10^{-59}	10^{-28}
				Knockoff Nets [30]	53.61	21.34	10^{-22}	10^{-15}	10^{-7}
				AugSteal [36]	86.68	6.08	10^{-1}	10^{-99}	10^{98}
				D-DAE [69]	84.26	18.93	10^{-20}	10^{-72}	10^{52}
				MEBooster [70]	85.20	23.50	10^{-28}	10^{-65}	10^{-37}
				SSLExtraction (Ours)	88.74	**5.50**	$\mathbf{10^{-1}}$	$\mathbf{10^{-105}}$	$\mathbf{10^{103}}$
MEA-Defender [71]	86.08	100.00	1.40 ± 1.14	Retraining [29]	81.26	45.40	10^{-16}	10^{-15}	10^{-2}
				Knockoff Nets [30]	82.70	57.93	10^{-29}	10^{-4}	10^{-25}
				AugSteal [36]	82.47	27.50	10^{-3}	10^{-46}	10^{43}
				D-DAE [69]	86.08	52.37	10^{-27}	10^{-10}	10^{-17}
				MEBooster [70]	85.51	67.72	10^{-28}	10^{-20}	10^{-8}
				SSLExtraction (Ours)	87.47	**2.33**	$\mathbf{10^{-1}}$	$\mathbf{10^{-107}}$	$\mathbf{10^{106}}$

tions identical. Additional implementation details are provided in Appendix A.

Evaluation with Multiple Benign Models. To obtain a more reliable estimate of watermark similarity, we extend the p-value-based evaluation by using multiple benign models rather than a single one. We train five benign models with diverse architectures (ResNet-50, VGG16, AlexNet, ViT, and Swin) under the same training protocol, and compute WSR for each model independently. The reported score is the mean and standard deviation across these five benign models. This multi-reference evaluation reduces variance introduced by model-specific behaviors and provides a more robust measure of the surrogate model's watermark alignment with benign models.

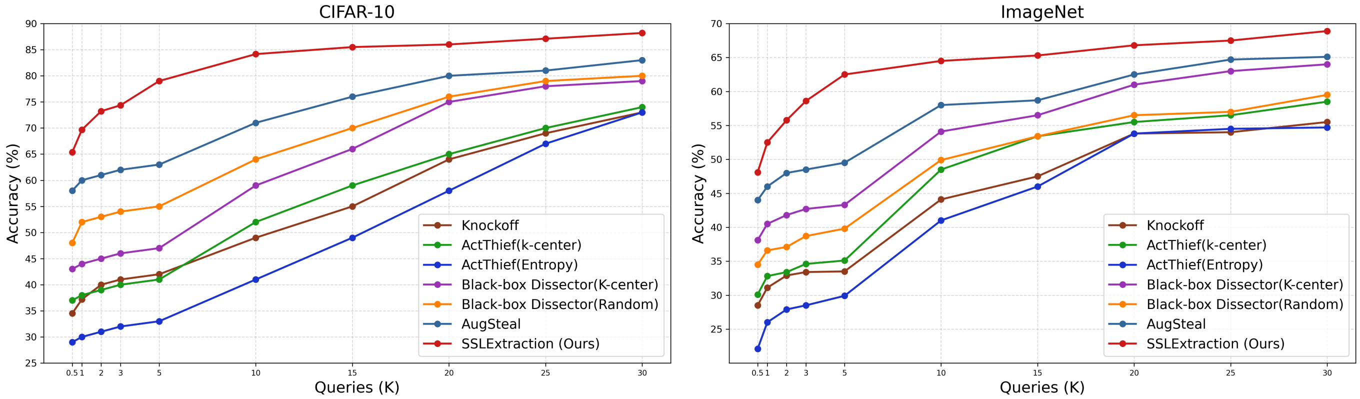
B. Results on Watermark Removal

Table I demonstrates that our method achieves the best watermark removal performance compared to other baseline removal attacks on CIFAR-10. A similar trend is observed on ImageNet, as shown in Table IX in Appendix A. Specifically, our approach attains the lowest WSR. It also achieves the highest p-value$_w$, suggesting that the surrogate model is statistically independent of the watermarked model in hypothesis testing and exhibits the greatest absolute distance from it.

We note that surrogate models rarely exceed the accuracy of the watermarked models. Within this limit, accuracy primarily relies on including more representative samples. This can be a challenge in watermark removal, as normal samples can be excluded while avoiding trigger samples, which in turn sacrifice accuracy. Our method mitigates this issue by strategically making the trigger samples as outliers in SSL feature space and highly separable from normal samples. Thus, the surrogate model simultaneously attains high accuracy and low WSR as shown in Table I. Specifically, our method leads to accuracy of 68-71% (ImageNet) and 84-89% (CIFAR-10), which are much reasonably close to watermarked model accuracy 70.06-74.25% (ImageNet) in Table IX (in Appendix A) and 86.08-87.81% (CIFAR10) in Table I respectively. Moreover, our method is particularly effective at achieving high accuracy with fewer queries (reflected in Fig. 4), and beyond 5K-10K queries, the gain becomes marginal since the surrogate approaches the watermarked model's accuracy limit.

For ImageNet extraction task, as shown in Table IX in Appendix A, although the encoder is pretrained on CIFAR-10, it still learns generic low-level and mid-level features that transfer across datasets. And the encoder only needs to produce a feature space in which samples with different characteristics can be meaningfully separated; it does not need to encode the full ImageNet distribution at this stage. After sample selection, we append a two-layer MLP to the encoder, and the victim-provided labels are then used to train the entire surrogate model. Through this supervised training process, both the encoder and the MLP progressively adapt and acquire the richer ImageNet-level representations. Consequently, a CIFAR-10-pretrained encoder is sufficient to bootstrap the sampling process while still enabling high accuracy.

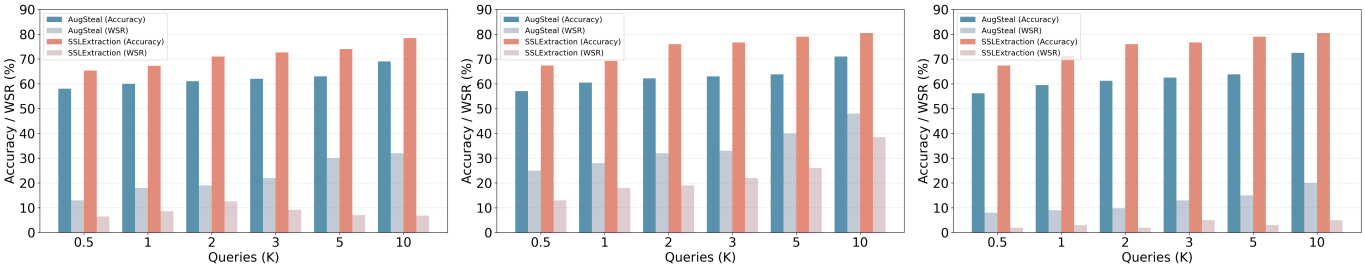
Benefiting from learning the inherent features of inputs, our method effectively **removes various types of watermarks**, including backdoor-based watermarks (EWE [21]), randomly assigned input-output pairs (Margin-based [25]), and deterministically selected input-output pairs (MAT [27]). Our approach also achieves strong removal performance against the state-of-



(a)

(b)

Fig. 4: Accuracy curves of surrogate models on CIFAR-10 and ImageNet under various model extraction methods.



(a) Margin-based Watermarking

(b) MAT

(c) MEA-Defender

Fig. 5: Accuracy and WSR of surrogate models on CIFAR-10 under various extraction attacks and watermarking schemes.

the-art method MEA-Defender [71] and successfully captures the intrinsic features, making the surrogate model more aligned with the benign model.

While our method does not always reduce the WSR to absolute zero, we note that perfectly distinctive watermarks have not been demonstrated by any prior watermarking scheme. Consistent with prior work [71], benign models naturally exhibit non-zero WSR due to accidental trigger activations. Following standard practice in the watermarking literature, which considers WSR $\leq$ 30% as successful unwatermarking [71], our method achieves a substantially lower WSR (Table I), outperforming all existing baselines. Although a small fraction of triggers may still activate, the reduction is sufficient to invalidate practical ownership claims under existing verification protocols.

Table I also demonstrates that **our proposed metric** r provides a more precise evaluation of the watermarking task and better mitigates false ownership claims compared to p-value. Notably, MAT exhibits the highest post-extraction WSR (44.49%) and the lowest p-value$_w$ among all watermarking methods. This correlation arises because both p-value$_w$ and WSR measure the similarity between the surrogate and watermarked models. Despite MAT yielding the highest WSR among all attacks, the surrogate model aligns more closely with the benign model (WSR = 45.40%), revealing an inconsistency. This motivates the introduction of a more reliable metric r for distinction. The surrogate model under MAT yields $r = 10^{98}$, indicating that it is relatively closer to the benign model. This further confirms that our method effectively removes the watermark despite the high WSR.

We further test prior extraction methods that were originally designed for unwatermarked models in the watermark removal setting. The results indicate that when queries are issued over the entire training set, AugSteal [36] fails to suppress the watermark. This limitation arises because the method is tailored to learn the victim's decision boundaries, which overlap with regions containing watermark information, thereby yielding a high WSR. In contrast, our framework leverages the intrinsic feature distribution of the inputs, naturally avoiding watermark-specific patterns. We also examine how different query budgets affect watermark retention, as illustrated in Figure 5, with further analysis in Section IV-C.

C. Results on Data Reduction

Fig. 4 presents the results of our method compared to other approaches for data reduction on the CIFAR-10 and ImageNet datasets. We obtain hard-label outputs by querying a watermarked model [25] and use these outputs to train both our linear classifier and other baseline models. The test accuracy

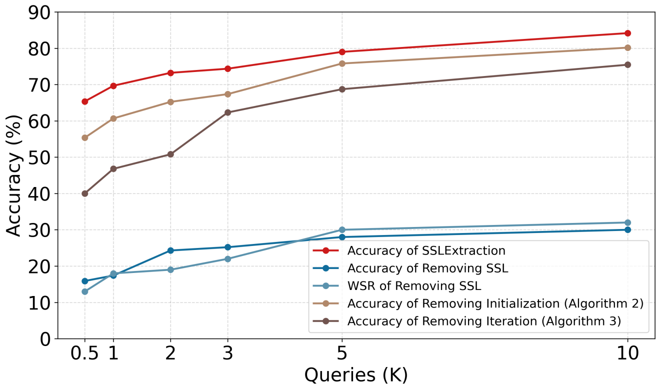


Fig. 6: Ablation study. Impact of key components on surrogate model's accuracy in our model extraction attack.

is recorded as the evaluation metric. We set the number of iterations T in Algorithm 1 to 100, which requires only a few minutes to produce an approximate solution to Problem (2).

To ensure a fair comparison with existing methods, we adopt the same experimental setup by training the self-supervised encoder on an out-of-distribution (OOD) dataset. Specifically, for attacks targeting CIFAR-10 classification models, we use an encoder trained on ImageNet, while for attacks against ImageNet models, we employ an encoder trained on CIFAR-10. After obtaining predictions from the watermarked victim model, we fine-tune the entire model while simultaneously training a linear classifier. In addition, Table VI shows that using in-distribution data for self-supervised training yields higher accuracy due to better alignment with the target task.

Experimental results show that our method requires significantly fewer queries than baselines. Specifically, our method achieves 70% accuracy with only 500 queries, whereas other methods require over 10K queries to reach this performance. This indicates that our algorithm produces a rather sharp estimate. It benefits from the random walk strategy on the lattice, which enables rapid convergence to a locally optimal solution. Our approach selects sufficiently diverse samples, allowing for near-complete coverage of the feature space. In contrast, other methods precisely mimic decision boundaries [36] or search within the pixel space [34], which leads to redundant queries.

Furthermore, our method not only achieves high query efficiency but also removes watermark information simultaneously, as shown in Fig. 5. The results show that the baseline method AugSteal [36] inadvertently captures and retains watermark information. In contrast, our method learns the inherent features, effectively identifying watermarked samples as outliers and mitigating their influence on surrogate model.

D. Ablation Study

We perform the ablation study by querying a watermarked model trained using the margin-based method [25] on CIFAR-10, which achieves 87.81% accuracy and a watermark success rate of 100.00%. The encoder is trained on an OOD dataset (ImageNet), and all queries are also sampled from the same OOD source. Our method comprises three key components: (1) self-supervised learning for feature extraction, (2) initialization, and (3) random walk strategy for iterative optimization. Experimental results, shown in Fig. 6, indicate that WSR remains consistently low (around 8%) in all configurations except when removing SSL. Therefore, we report the WSR only for that particular case and focus the figure on highlighting the impact of each component on data reduction.

Impacts of Removing SSL. When self-supervised learning is removed, the surrogate model is trained entirely from random initialization without any pretrained or frozen encoder, and Algorithm 1 can only select dissimilar inputs in the pixel space. The surrogate model achieves low accuracy and relatively high WSR for all query budgets, as the sampled points remain dispersed in the pixel space and tend to learn watermark-related features rather than capturing intrinsic representations.

Impacts of Initialization (Algorithm 2). When the greedy-based initialization is removed, we adopt a random initialization strategy. As shown in Fig. 6, the performance under random initialization is only slightly inferior to that of greedy initialization. This is because our method incorporates an iterative random walk process following initialization, which allows the solution to approximate a locally optimal solution to Problem (2). We further analyze the effect of the number of iterations in the next section, with results presented in Table III. The results indicate that greedy initialization enables faster convergence, reaching stability within a small number of iterations, whereas the random initialization strategy requires over 100 iterations to achieve a comparable performance. Thus, greedy initialization improves computational efficiency and reduces resource consumption.

Impacts of Random Walk (Algorithm 3). When the random walk algorithm is removed, we directly query the victim model using the initialization results. As shown in Fig. 6, the accuracy drops significantly when the query budget is below 1K. This occurs because, even with greedy initialization, the first randomly selected sample can influence the subsequent queries, limiting the initialization to fully exploring and covering the entire feature space. However, when the query budget exceeds 5K, the accuracy degradation becomes less severe. This is because a larger query budget increases the likelihood of sampling from previously uncovered regions of the feature space, thereby expanding coverage and reducing the dependence on the random walk algorithm.

E. Other Impacts

Impacts of Different SSL Frameworks. We evaluate our method using three self-supervised learning frameworks: SimCLR [43], MoCo v2 [44], and BYOL [45], with results presented in Table II. The differences among them are minimal, mainly in slight variations in downstream task accuracy. This is expected, as they are representative contrastive learning algorithms capable of extracting intrinsic feature representations.

Impacts of Random Walk Iterations. In our main experiments, the number of iterations T in Algorithm 3 is set to 100. Here, we evaluate the impact of T on performance. As shown

TABLE II: Impacts of different SSL frameworks on CIFAR-10 dataset.

Victim Models			Surrogate Models									
Acc. (%)	WSR (%)	SSL frameworks	500 Query		1K Query		2K Query		3K Query		5K Query	
			Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)
83.57	100.00	SimCLR [43]	65.36	6.45	67.24	8.61	70.97	12.56	72.65	9.15	73.98	7.01
		MoCo v2 [44]	58.12	8.11	61.55	9.65	67.59	13.91	69.74	8.61	69.55	10.41
		BYOL [45]	72.15	2.12	75.82	3.29	77.67	2.79	81.41	5.62	83.87	3.36

TABLE III: Impacts of random walk iterations.

Queries	Initialization Strategy	Iteration				
		10	20	30	50	100
500	Greedy (Algorithm 2)	64.26	65.43	65.49	65.12	65.36
	Random	61.91	63.44	64.15	64.44	65.17
1K	Greedy (Algorithm 2)	66.85	66.92	67.40	67.30	67.24
	Random	64.91	65.30	65.84	66.46	67.05
2K	Greedy (Algorithm 2)	69.43	69.79	70.50	70.12	70.97
	Random	67.82	68.90	70.08	70.42	70.61
3K	Greedy (Algorithm 2)	71.70	72.54	72.08	72.63	72.65
	Random	70.48	71.57	72.45	72.04	72.68
5K	Greedy (Algorithm 2)	73.01	73.73	73.88	74.06	73.98
	Random	72.46	73.58	73.74	73.38	73.33

in Table III, when the query budget is small, approximately 100 iterations are required for convergence. However, with a larger query budget, convergence is achieved with fewer iterations since the initial query set already provides sufficient coverage of the feature space and only minor refinements are needed to reach a local optimum. Although random initialization requires more iterations to reach convergence, increasing the query budget mitigates this to some extent. Compared to random initialization, Algorithm 3 consistently converges faster across all query budgets, demonstrating its efficiency.

Impacts of Victim-Surrogate Cross-Architecture Settings. In our threat model, the architecture of the victim model is kept confidential for intellectual property protection. To assess whether our method depends on architectural similarity between victim and surrogate models, we conduct a cross-architecture study using five distinct architectures, including three CNN-based models (VGG16 [42], AlexNet [75], and ResNet-50 [1]) and two transformer-based models (ViT [2] and Swin [76]). As shown in Table V and Table IV, our method achieves comparable extraction performance across all architecture pairs. Because the victim models in our threat model return only *hard labels* rather than *soft predictions*, the queries produce consistent supervision signals regardless of the victim's architecture. Consequently, surrogate models trained on these responses achieve nearly identical accuracy, demonstrating that our approach is robust to mismatched architectures and does not rely on shared inductive biases or feature geometry between victim and surrogate models, enabling the attacker to choose surrogate architectures freely.

Impacts of Different Training Datasets for SSL. We evaluate the influence by training the encoder using both in-distribution and OOD datasets. As shown in Table VI, when in-distribution data is available to adversary, the accuracy of the extracted model reaches 83.15% accuracy with only 500 queries, alongside a reduced WSR. This highlights the significant role of distribution alignment in improving the effectiveness of model extraction and further reducing WSR, underscoring the critical need for data privacy preservation in real-world deployments.

V. Discussion

Black-box DNN Watermarks and Backdoors. Black-box watermarks are sometimes referred to as backdoor-based in prior work [78]. However, they differ significantly from standard DNN backdoors in terms of data patterns and target label settings, especially in non-fixed-class scenarios. As a result, removal techniques adapted from backdoor defenses, are often ineffective against black-box watermarks [68].

Applications for p-Dispersion-Sum Problem. The maximum dispersion problem has a wide range of practical applications, such as optimizing facility location [79] to prevent facility destruction in military defense and reduce competition in business planning. Beyond physical distance, it also extends to genetics, promoting genetic diversity [80], and social diversity in workplace environments [81]. Additionally, it has been applied to optimize seating for COVID-19 social distancing [82].

In this work, we extend the concept of dispersion to feature space to guide feature selection in our extraction attack. By ensuring broad coverage of the feature space, our approach minimizes redundancy and reduces the query budget.

Algorithm for p-Dispersion-Sum Problem. In Section III-C, we propose a biased random walk algorithm to solve the p-dispersion-sum problem. Although a variety of exact and reformulation-based techniques have been studied (including relaxations, semidefinite or integer linear reformulations, and concave optimization) these approaches share the limitation of rapidly escalating computational cost. Numerical results indicate that these methods can efficiently handle instances with $n = 80$ in 60 seconds but struggle for $n = 100$. Given these limitations, we design a biased random walk algorithm on a lattice, providing an efficient and practical solution.

Results on White-Box Watermarks. Beyond black-box settings, our method also proves effective against white-box watermarks. In evaluations on three representative white-box schemes [9], [83], [10], it reduces watermark accuracy to around 10%, indicating successful removal (Table VII).

Results on Defense Mechanisms. For methods that perturb confidence scores [85], [86], [70], [87], they preserve hard-label outputs to maintain utility and any effective modification of the logits would typically cause surrogate models to suffer noticeable accuracy degradation. However, since our threat model assumes access to hard labels only, our method is not affected by these defenses and maintains high accuracy as

TABLE IV: Impacts of different victim architectures on CIFAR-10 MEA-Defender watermarked models.

Victim Models			Surrogate Models									
			500 Query		1K Query		2K Query		3K Query		5K Query	
Architectures	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)
VGG-like	81.07	100.00	63.28	9.68	68.57	6.23	72.14	7.29	72.83	8.48	74.09	8.74
AlexNet	80.87	100.00	63.45	5.56	67.36	12.79	71.41	10.43	72.40	7.56	73.33	5.13
ResNet-50	86.08	100.00	67.36	2.62	68.24	3.66	71.97	2.09	73.98	3.10	74.98	5.24

TABLE V: Results for cross-architecture model extraction on CIFAR-10 Margin-based watermarked models.

Victim Models			Surrogate Models		
Architectures	Acc. (%)	WSR (%)	Architectures	Acc. (%)	WSR (%)
ResNet-50	87.81	100.00	ResNet-50	88.02	5.39
			AlexNet	84.27	6.12
			VGG	85.91	5.84
			ViT	86.34	5.51
			Swin	86.72	5.63
AlexNet	88.57	100.00	ResNet-50	86.91	5.47
			AlexNet	88.17	6.02
			VGG	85.45	5.78
			ViT	86.12	5.66
			Swin	87.47	5.71
VGG	85.03	100.00	ResNet-50	87.42	5.41
			AlexNet	83.95	6.05
			VGG	85.27	5.62
			ViT	85.76	5.53
			Swin	86.01	5.59
ViT	72.65	100.00	ResNet-50	85.38	5.44
			AlexNet	80.71	6.17
			VGG	83.62	5.89
			ViT	82.41	5.98
			Swin	84.75	5.66
Swin	88.11	100.00	ResNet-50	86.12	5.52
			AlexNet	88.03	6.21
			VGG	88.91	5.88
			ViT	84.23	5.74
			Swin	86.57	6.03

shown in Table XI in Appendix A. Then, for defenses that relabel predictions [84], our method operates in the intrinsic feature space, making label manipulation ineffective. As for query-filtering defenses targeting statistical anomalies [33], our queries are non-adversarial and distributionally indistinguishable from natural samples.

Results on non-vision modalities. To verify the generalization of our approach beyond computer vision, we extended the experiments to text [71], audio [71], and SSL-encoder [88], [71], [89] watermarking scenarios. Since most existing watermarking methods rely on outlier-based trigger designs, our SSL-based approach can effectively identify and filter out such outliers in the representation space, enabling consistent watermark removal across different modalities. Table VIII reports our cross-modal experimental results, where the MAD (Median Absolute Deviation) [89] quantifies the outlierness in the output-entropy distribution [89], and the WR [88] denotes the fraction of queries whose outputs match the watermark trigger pattern [88]. Our method consistently achieves the high task accuracy and low watermark verification metrics across different modalities (text, audio and encoder), indicating that the surrogate models exhibit watermark behaviors close to those of the benign models.

Results on Victim-Class Absence. To examine how missing victim classes affect SSL training, we construct training sets by removing all CIFAR-10 samples from 1, 2, and 5 randomly selected classes, and then extract the watermarked CIFAR-10 victim model [25]. We report the average performance over 10 independent trials. As shown in Table XII in Appendix A, our extraction attack remains effective and consistently removes watermarks across all settings. We further evaluate class-specific absence by removing all dog images from CIFAR-10 to extract the watermarked CIFAR-10 victim model [25]. The overall surrogate accuracy drops only moderately, and the accuracy on the dog class remains high (82%), despite the complete absence of dog samples during SSL pretraining. A similar trend is observed when removing all dog-related classes from ImageNet for training set to extract the watermarked ImageNet victim model [25]. These results indicate that our method does not require OOD coverage of all victim classes; instead, it remains robust by leveraging sufficiently dispersed queries to learn transferable image representations. We additionally explored whether generative AI can compensate for missing classes. By removing five CIFAR-10 classes and generating 1,000 synthetic images per removed class with Stable Diffusion, the resulting 84.51% accuracy and 6.52% WSR remain close to the 5-Class Missing setting in Table XII in Appendix A, showing only marginal improvement. A more comprehensive investigation into using generative AI to create training samples can be explored as future work.

Adversary Capabilities. In this work, we impose no constraints on the adversary's capabilities, assuming unlimited computational resources and access to unlabeled inputs. Under this idealized setting, our study demonstrates that with the SSL framework, an adversary can obtain a near-perfect encoder and adapt to downstream tasks with minimal queries. This highlights that model extraction attacks should not be analyzed under the assumption of unlimited computational power, not even at an exponential scale. If such computational resources were available, the adversary could effortlessly train a high-performing SSL encoder and efficiently solve Optimization Problem (2), minimizing query budget.

Rather than assuming constraints on the adversary's capabilities, protecting the privacy of proprietary data is a more critical concern for defenders. As shown in Table VI, when the adversary acquires in-distribution knowledge, they can achieve a near-converged accuracy of 83.15% with only 100 queries, while also significantly reducing the WSR. These results underscore the importance of safeguarding training data, as access to such information can significantly amplify the effectiveness of model extraction and watermark removal.

TABLE VI: Impacts of different training datasets for SSL in extracting watermarked model for CIFAR-10 classification.

SSL Training Datasets	Surrogate Models									
	500 Query		1K Query		2K Query		3K Query		5K Query	
	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)
STL10 [77]	65.36	8.18	67.24	8.27	70.97	12.15	72.65	9.06	73.98	10.50
ImageNet [73]	67.15	6.45	69.61	6.61	73.74	10.56	74.86	9.15	77.98	7.01
CIFAR-10 [72]	83.15	2.85	83.82	3.04	84.65	2.63	84.41	5.24	84.87	3.51

TABLE VII: Results of our method against non-black-box watermarking schemes on the CIFAR-10 dataset.

Victim Models			Surrogate Models									
Defense Methods	Acc. (%)	WSR (%)	500 Query		1K Query		2K Query		3K Query		5K Query	
			Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)
Uchida [9]	90.19	100.00	70.68	10.21	71.93	12.24	76.35	11.98	77.29	12.91	81.53	11.74
DeepMarks [83]	91.13	100.00	71.18	7.71	72.39	5.74	76.88	6.53	77.82	7.43	82.03	6.28
DeepSigns [10]	91.22	100.00	71.13	11.65	72.47	6.78	76.80	8.44	77.75	7.48	82.01	8.23
DAWN [84]	92.09	99.46	71.59	12.03	72.82	14.12	77.41	12.85	78.37	13.74	82.51	14.63

TABLE VIII: Results for watermark removal across non-vision modalities including text, audio, and SSL-encoder tasks.

Tasks	Watermarking Methods	Datasets	Victim Models		Benign Models	Surrogate Models	
			Acc. (%)	Verification Metric	Verification Metric	Acc. (%)	Verification Metric
Text	MEA-Defender [71]	AG News	88.15	WSR = 100.00%	WSR = 1.19%	90.75	WSR = 2.54%
Audio	MEA-Defender [71]	Speech Commands	82.17	WSR = 100.00%	WSR = 0.92%	80.50	WSR = 3.86%
Encoder	SSLGuard [88]	CIFAR-10	76.50	WR = 100.00%	WR = 0.04%	81.78	WR = 0.81%
	MEA-Defender [71]		75.14	WSR = 100.00%	WSR = 0.38%	82.90	WSR = 1.26%
	SSL-WM [89]		84.33	MAD = 99.45	MAD = −0.92	87.15	MAD = −0.30

Limitation. Our metric r requires access to benign models in order to quantify the statistical deviation between benign and surrogate behaviors. For watermarking schemes that embed triggers via a fine-tuning stage (such as EWE [21] and MAT [27]), the benign model naturally exists as the checkpoint prior to watermark insertion, and thus no additional training cost is incurred. However, for watermarking methods that train the watermarked model from scratch without a benign checkpoint (such as Margin-based [25] and MEA-Defender [71]), an additional benign model must be trained to compute r. While this extra cost is unavoidable for such schemes, we emphasize that it is incurred only once by the model owner.

VI. Related Work

Most of the watermark removal works focus on white-box settings. Pruning-based attacks [90], [9] remove a large number of weights or neurons. Finetuning-based attacks [91], [92], [93], [47], [94] involve further training the target model with a smaller learning rate and dataset. Unlearning-based attacks [78], [95] identify watermark patterns and remove them using unlearning techniques.

In this paper, we consider a more realistic black-box setting where the model architecture and parameters are kept confidential, and only limited API access is available. Retraining [29] is oftentimes considered the first model extraction. Sampling-based approaches, such as Knockoff Nets' random/adaptive selection [30] and reservoir sampling [96], as well as SwiftThief's rare-class-prioritized sampling [97], all aim to identify more informative public data to improve extraction efficiency. More recent approaches include detection-and-recovery attacks [69], [98], training-optimization-based methods like MEBooster [70]. However, existing methods require a large number of queries to obtain functionally equivalent surrogate models, yet often retain the watermark patterns.

Model extraction attacks typically require many queries, so data reduction seeks to lower query costs while preserving accuracy. Early works like Jacobian-based augmentation [33] and DRMI [34] select informative samples but rely on data similar to the target model, limiting practicality. Copycat CNN [99] and Knockoff Nets [30] instead leverage public datasets and adaptive sampling to avoid dependence on proprietary data. AugSteal [36] further enhances query efficiency through public data filtering, adaptive querying, and augmentation. Data-free model extraction methods synthesize queries using generative models [100], [101], [102] to steal the victim model without relying on any real data. However, these methods often miss task-intrinsic features, leading to redundant queries and unintended learning of watermark patterns.

VII. Conclusion

In this paper, we present SSLExtraction, a black-box model extraction framework that leverages SSL for data reduction and watermark removal. We first extract intrinsic and watermark-independent representations. Then, we formalize the query sample selection process as a p-dispersion-sum optimization problem. After analyzing the computational complexity of this problem, we propose a high-dimensional random walk-based approximation algorithm, which significantly reduces the query budget while maintaining high model extraction accuracy and effective watermark removal. Furthermore, we emphasize that the goal of watermark task is to determine whether a suspicious model is a benign model or a stolen one. To this end, we introduce a new evaluation metric that leverages hypothesis testing to quantify the relative distance from a suspicious model to both a watermarked model and a benign model, determining which it more closely resembles. Extensive experiments across various watermarking defenses and extraction attacks demonstrate that SSLExtraction not only achieves strong extraction performance but also effectively removes watermarks, outperforming existing baselines.

ACKNOWLEDGMENT

Hao Luan, Xue Tan, and Ping Chen were funded in part by the National Key R&D Program of China under Grant 2022YFB3104300.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE CVPR*, 2016, pp. 770–778.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, "Medical image classification using synergic deep learning," *Medical image analysis*, vol. 54, 2019.

[6] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE intelligent systems*, vol. 24, no. 2, pp. 8–12, 2009.

[7] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

[8] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Watermarking techniques for intellectual property protection," in *Proceedings of the 35th annual Design Automation Conference*, 1998, pp. 776–781.

[9] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *ICMR*, 2017, pp. 269–277.

[10] B. Darvish Rouhani, H. Chen, and F. Koushanfar, "Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks," in *ASPLOS*, 2019, pp. 485–497.

[11] M. Kuribayashi, T. Tanaka, S. Suzuki, T. Yasui, and N. Funabiki, "White-box watermarking scheme for fully-connected layers in fine-tuning model," in *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021, pp. 165–170.

[12] D. Mehta, N. Mondol, F. Farahmandi, and M. Tehranipoor, "Aime: Watermarking ai models by leveraging errors," in *DATE*. IEEE, 2022.

[13] 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*, 2018, pp. 1615–1631.

[14] 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 the 2018 on Asia CCS*, 2018.

[15] Z. Li, C. Hu, Y. Zhang, and S. Guo, "How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of dnn," in *Proceedings of the 35th annual computer security applications conference*, 2019, pp. 126–137.

[16] R. Namba and J. Sakuma, "Robust watermarking of neural network with exponential weighting," in *ACM Asia CCS*, 2019.

[17] J. Zhang, D. Chen, J. Liao, H. Fang, W. Zhang, W. Zhou, H. Cui, and N. Yu, "Model watermarking for image processing networks," in *AAAI*, vol. 34, no. 07, 2020, pp. 12 805–12 812.

[18] J. Zhang, D. Chen, J. Liao, W. Zhang, G. Hua, and N. Yu, "Passport-aware normalization for deep model protection," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 619–22 628, 2020.

[19] X. Chen, T. Chen, Z. Zhang, and Z. Wang, "You are caught stealing my winning lottery ticket! making a lottery ticket claim its ownership," *Advances in neural information processing systems*, vol. 34, 2021.

[20] P. Yang, Y. Lao, and P. Li, "Robust watermarking for deep neural networks via bi-level optimization," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 841–14 850.

[21] H. Jia, C. A. Choquette-Choo, V. Chandrasekaran, and N. Papernot, "Entangled watermarks as a defense against model extraction," in *30th USENIX security*, 2021.

[22] P. Maini, M. Yaghini, and N. Papernot, "Dataset inference: Ownership resolution in machine learning," *arXiv preprint arXiv:2104.10706*.

[23] Y. Li, L. Zhu, X. Jia, Y. Jiang, S.-T. Xia, and X. Cao, "Defending against model stealing via verifying embedded external features," in *AAAI*, vol. 36, no. 2, 2022, pp. 1464–1472.

[24] A. Bansal, P.-y. Chiang, M. J. Curry, R. Jain, C. Wigington, V. Manjunatha, J. P. Dickerson, and T. Goldstein, "Certified neural network watermarks with randomized smoothing," in *ICML*. PMLR, 2022.

[25] B. Kim, S. Lee, S. Lee, S. Son, and S. J. Hwang, "Margin-based neural network watermarking," in *ICML*. PMLR, 2023, pp. 16 696–16 711.

[26] S. Yu, J. Hong, H. Zhang, H. Wang, Z. Wang, and J. Zhou, "Safe and robust watermark injection with a single ood image," *arXiv preprint arXiv:2309.01786*, 2023.

[27] Y. Li, S. K. Maharana, and Y. Guo, "Not just change the labels, learn the features: Watermarking deep neural networks with multi-view data," *arXiv preprint arXiv:2403.10663*, 2024.

[28] M. Pautov, N. Bogdanov, S. Pyatkin, O. Rogov, and I. Oseledets, "Probabilistically robust watermarking of neural networks," *arXiv preprint arXiv:2401.08261*, 2024.

[29] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {APIs}," in *25th USENIX security*, 2016, pp. 601–618.

[30] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *IEEE/CVF CVPR*, 2019.

[31] S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy, "A framework for the extraction of deep neural networks by leveraging public data," *arXiv preprint arXiv:1905.09165*, 2019.

[32] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia CCS*, 2017, pp. 506–519.

[33] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: protecting against dnn model stealing attacks," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 512–527.

[34] Y. He, G. Meng, K. Chen, X. Hu, and J. He, "{DRMI}: A dataset reduction technology based on mutual information for black-box attacks," in *30th USENIX security*, 2021, pp. 1901–1918.

[35] S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy, "Activethief: Model extraction using active learning and unannotated public data," in *AAAI*, vol. 34, no. 01, 2020, pp. 865–872.

[36] L. Gao, W. Liu, K. Liu, and J. Wu, "Augsteal: Advancing model steal with data augmentation in active learning frameworks," *TIFS*, 2024.

[37] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[38] P. Munjal, N. Hayat, M. Hayat, J. Sourati, and S. Khan, "Towards robust and reproducible active learning using neural networks," in *Proceedings of the IEEE/CVF CVPR*, 2022, pp. 223–232.

[39] M. J. Kuby, "Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems," *Geographical Analysis*, vol. 19, no. 4, pp. 315–329, 1987.

[40] J. Liu, R. Zhang, S. Szyller, K. Ren, and N. Asokan, "False claims against model ownership resolution," in *33rd USENIX security*, 2024.

[41] Y. Wang, J. Li, H. Liu, Y. Wang, Y. Wu, F. Huang, and R. Ji, "Black-box dissector: Towards erasing-based hard-label model stealing attack," in *European conference on computer vision*. Springer, 2022.

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

[43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PmLR, 2020, pp. 1597–1607.

[44] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*.

[45] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Boot-

strap your own latent-a new approach to self-supervised learning," *NeurIPS*, vol. 33, 2020.

[46] M. Thomas and A. T. Joy, *Elements of information theory*. Wiley-Interscience, 2006.

[47] M. Shafieinejad, N. Lukas, J. Wang, X. Li, and F. Kerschbaum, "On the robustness of backdoor-based watermarking in deep neural networks," in *Proceedings of the 2021 ACM workshop on information hiding and multimedia security*, 2021, pp. 177–188.

[48] X. He, Z. Li, W. Xu, C. Cornelius, and Y. Zhang, "Membership-doctor: Comprehensive assessment of membership inference against machine learning models," *arXiv preprint arXiv:2208.10445*, 2022.

[49] X. He, H. Liu, N. Z. Gong, and Y. Zhang, "Semi-leak: Membership inference attacks against semi-supervised learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 365–381.

[50] X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, "Node-level membership inference attacks against graph neural networks," *arXiv preprint arXiv:2102.05429*, 2021.

[51] X. He and Y. Zhang, "Quantifying and mitigating privacy risks of contrastive learning," in *Proceedings of the 2021 ACM CCS*, 2021.

[52] Z. Li, Y. Liu, X. He, N. Yu, M. Backes, and Y. Zhang, "Auditing membership leakages of multi-exit networks," in *Proceedings of the 2022 ACM CCS*, 2022, pp. 1917–1931.

[53] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proceedings of the 2021 ACM CCS*, 2021, pp. 880–895.

[54] H. Liu, J. Jia, W. Qu, and N. Z. Gong, "Encodermi: Membership inference against pre-trained encoders in contrastive learning," in *Proceedings of the 2021 ACM CCS*, 2021, pp. 2081–2095.

[55] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[56] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE SP*, 2017.

[57] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th USENIX security*, 2021.

[58] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, "Badnl: Backdoor attacks against nlp models with semantic-preserving improvements," in *Proceedings of the 37th Annual Computer Security Applications Conference*, 2021, pp. 554–569.

[59] J. Jia, Y. Liu, and N. Z. Gong, "Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning," in *IEEE SP*, 2022.

[60] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *AAAI*, vol. 34, no. 07, 2020, pp. 11 957–11 965.

[61] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proceedings of the 2019 ACM CCS*, 2019.

[62] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.

[63] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[64] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[65] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.

[66] R. Rana and R. Singhal, "Chi-square test and its application in hypothesis testing," *Journal of the practice of cardiovascular sciences*, vol. 1, no. 1, pp. 69–71, 2015.

[67] N. Lukas, E. Jiang, X. Li, and F. Kerschbaum, "Sok: How robust is image classification deep neural network watermarking?" in *SP*, 2022.

[68] Y. Lu, W. Li, M. Zhang, X. Pan, and M. Yang, "Neural dehydration: Effective erasure of black-box watermarks from dnns with limited data," in *Proceedings of the 2024 on ACM CCS*, 2024, pp. 675–689.

[69] Y. Chen, R. Guan, X. Gong, J. Dong, and M. Xue, "D-dae: Defense-penetrating model extraction attacks," in *2023 IEEE SP*. IEEE, 2023.

[70] Y. Xiao, H. Hu, Q. Ye, L. Tang, Z. Liang, and H. Zheng, "Unlocking high-fidelity learning: Towards neuron-grained model extraction," *IEEE Transactions on Dependable and Secure Computing*, 2025.

[71] P. Lv, H. Ma, K. Chen, J. Zhou, S. Zhang, R. Liang, S. Zhu, P. Li, and Y. Zhang, "Mea-defender: a robust watermark against model extraction attack," in *2024 IEEE SP*. IEEE, 2024, pp. 2515–2533.

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

[73] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE CVPR*, 2009.

[74] Y. LeCun, "The mnist database of handwritten digits," *http://yann.lecun. com/exdb/mnist/*, 1998.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[76] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[77] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[78] W. Aiken, H. Kim, S. Woo, and J. Ryoo, "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks," *Computers & Security*, vol. 106, p. 102277, 2021.

[79] R. L. Church and R. S. Garfinkel, "Locating an obnoxious facility on a network," *Transportation science*, vol. 12, no. 2, pp. 107–118, 1978.

[80] W. Porter, K. Rawal, K. Rachie, H. Wien, and R. Williams, "Cowpea germplasm catalog no 1," *International institute of tropical agriculture, Ibadan, Nigeria*, 1975.

[81] M.-É. Roberge and R. Van Dick, "Recognizing the benefits of diversity: When and how does diversity increase group performance?" *Human Resource management review*, vol. 20, no. 4, pp. 295–308, 2010.

[82] R. Ferrero-Guillén, J. Díez-González, P. Verde, A. Martínez-Gutiérrez, J.-M. Alija-Pérez, and R. Álvarez, "Optimal chair location through a maximum diversity problem genetic algorithm optimization," in *International Work-Conference on Bioinformatics and Biomedical Engineering*. Springer, 2022, pp. 417–428.

[83] H. Chen, B. D. Rouhani, C. Fu, J. Zhao, and F. Koushanfar, "Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models," in *ICMR*, 2019, pp. 105–113.

[84] S. Szyller, B. G. Atli, S. Marchal, and N. Asokan, "Dawn: Dynamic adversarial watermarking of neural networks," in *Proceedings of the 29th ACM international conference on multimedia*, 2021.

[85] M. Tang, A. Dai, L. DiValentin, A. Ding, A. Hass, N. Z. Gong, Y. Chen *et al.*, "{ModelGuard}:{Information-Theoretic} defense against model extraction attacks," in *33rd USENIX security*, 2024, pp. 5305–5322.

[86] H. Chen, T. Zhu, L. Zhang, B. Liu, D. Wang, W. Zhou, and M. Xue, "Queen: Query unlearning against model extraction," *IEEE TIFS*, 2025.

[87] X. Gong, R. Wei, Z. Wang, Y. Sun, J. Peng, Y. Chen, and Q. Wang, "Beowulf: Mitigating model extraction attacks via reshaping decision regions," in *Proceedings of the 2024 on ACM CCS*, 2024.

[88] T. Cong, X. He, and Y. Zhang, "Sslguard: A watermarking scheme for self-supervised learning pre-trained encoders," in *2022 ACM CCS*.

[89] P. Lv, P. Li, S. Zhu, S. Zhang, K. Chen, R. Liang, C. Yue, F. Xiang, Y. Cai, H. Ma *et al.*, "Ssl-wm: A black-box watermarking approach for encoders pre-trained by self-supervised learning," *arXiv preprint arXiv:2209.03563*, 2022.

[90] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International symposium on research in attacks, intrusions, and defenses*. Springer, 2018, pp. 273–294.

[91] X. Chen, W. Wang, C. Bender, Y. Ding, R. Jia, B. Li, and D. Song, "Refit: a unified watermark removal framework for deep learning systems with limited data," in *ACM Asia CCS*, 2021.

[92] X. Chen, W. Wang, Y. Ding, C. Bender, R. Jia, B. Li, and D. Song, "Leveraging unlabeled data for watermark removal of deep neural networks," in *ICML workshop on Security and Privacy of Machine Learning*, 2019, pp. 1–6.

[93] X. Liu, F. Li, B. Wen, and Q. Li, "Removing backdoor-based watermarks in neural networks with limited data," in *2020 ICPR*.

[94] Q. Zhong, L. Y. Zhang, S. Hu, L. Gao, J. Zhang, and Y. Xiang, "Attention distraction: Watermark removal through continual learning with selective forgetting," in *2022 IEEE ICME*. IEEE, 2022, pp. 1–6.

[95] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE SP*. IEEE, 2019, pp. 707–723.

[96] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[97] J. Lee, S. Han, and S. Lee, "Swiftthief: enhancing query efficiency of model stealing by contrastive learning," in *Proc. 33rd Int. Joint Conf. Artif. Intell.* Aug, 2024, pp. 422–430.

[98] X. Gong, S. Li, Y. Chen, M. Li, R. Wei, Q. Wang, and K.-Y. Lam, "Augmenting model extraction attacks against disruption-based defenses," *IEEE TIFS*, 2024.

[99] J. R. Correia-Silva, R. F. Berriel, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data," in *IJCNN*. IEEE, 2018.

[100] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot, "Data-free model extraction," in *IEEE/CVF CVPR*, 2021, pp. 4771–4780.

[101] S. Kariyappa, A. Prakash, and M. K. Qureshi, "Maze: Data-free model stealing attack using zeroth-order gradient estimation," in *Proceedings of the IEEE/CVF CVPR*, 2021, pp. 13 814–13 823.

[102] S. Sanyal, S. Addepalli, and R. V. Babu, "Towards data-free model stealing in a hard label setting," in *IEEE/CVF CVPR*, 2022.

[103] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.

[104] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *IEEE/CVF CVPR*, 2019.

[105] R. M. Karp, "Reducibility among combinatorial problems," in *50 Years of Integer Programming 1958-2008: from the Early Years to the State-of-the-Art*. Springer, 2009, pp. 219–241.

Algorithm 1: Data Reduction

```
Input  : Feature vectors $\{h_i\}_{i=1}^n$, number of queries $p$,
         number of iterations $T$;
Output: Binary decision vector $\{b_i\}_{i=1}^n$ where
         $\sum_{i=1}^n b_i = p$;
$\{b_i\}_{i=1}^n$ = greedy_initialization($h_i$);
for all $j = 1, \cdots, T$ do
    $\{b_i\}_{i=1}^n$ = random_walk ($\{b_i\}_{i=1}^n, p, h_i$);
return $\{b_i\}_{i=1}^n$;
```

Algorithm 2: Greedy Initialization

```
Input  : Feature vectors $\{h_i\}_{i=1}^n$, number of queries $p$;
Output: Binary decision vector $\{b_i\}_{i=1}^n$ where
         $\sum_{i=1}^n b_i = p$;
$\{b_i\}_{i=1}^n = 0$;
$j = \text{RandomSelect}(1, n)$;
$b_j = 1$;
for all $j = 2, \cdots, p$ do
    for all $k = 1, \cdots, n$ do
        if $b_k = 0$ then
            $d_k = \sum_{i=1}^n b_i \cdot \|h_i - h_k\|$;
        else
            $d_k = 0$;
    $\ell = \arg\max_k d_k$;
    $b_\ell = 1$;
return $\{b_i\}_{i=1}^n$;
```

Algorithm 3: Random Walk Iteration

```
Input  : Feature vectors $\{h_i\}_{i=1}^n$, number of queries $p$,
         binary decision vector $\{b_i\}_{i=1}^n$;
Output: Binary decision vector $\{b_i\}_{i=1}^n$ where
         $\sum_{i=1}^n b_i = p$;
$j = \text{RandomSelect}(1, n)$;
while $b_j = 1$ do
    $j = \text{RandomSelect}(1, n)$;
$b_j = 1$;
for all $k = 1, \cdots, n$ do
    if $b_k = 1$ then
        $d_k = \frac{1}{2} \sum_{i \neq k} \sum_{j \neq k} b_i b_j \cdot \|h_i - h_j\|$;
    else
        $d_k = 0$;
$\ell = \arg\max_k d_k$;
$b_\ell = 0$;
return $\{b_i\}_{i=1}^n$;
```

Algorithm 4: Ownership Verification

```
Input  : Watermarked model $\tilde{M}$, suspicious model $\hat{M}$,
         benign model $M_b$, trigger samples $\{x_i\}_{i=1}^m$;
Output: Ratio $r$;
$y_i^w = \tilde{M}(x_i)$;
$y_i^s = \hat{M}(x_i)$;
$y_i^b = M_b(x_i)$;
p-value$_w = \chi^2 -$ Text $(\{y_i^s\}_{i=1}^m, \{y_i^w\}_{i=1}^m)$;
p-value$_b = \chi^2 -$ Text $(\{y_i^s\}_{i=1}^m, \{y_i^b\}_{i=1}^m)$;
$r = \frac{\text{p-value}_w}{\text{p-value}_b}$;
return $r$;
```

Appendix A
Additional Experimental Details

A. Experimental Setup

Watermarking Methods. We evaluate all extraction methods against a diverse set of representative watermarking schemes encompassing OOD triggers, in-distribution sampling, boundary-based selection, backdoor techniques, and composite-pattern watermarking. Specifically, we use four representative watermarking methods: Margin-based Watermarking [25], Multi-View Data (MAT) [27], Entangled Watermark Embedding (EWE) [21] and MEA-Defender [71]. All baselines are evaluated using their official public implementations.

Baseline Attacks. We compare our method with Retraining [29], Knockoff Nets [30] and AugSteal [36] for watermark removal. And we compare our method with the ActiveThief [35] and Black-box Dissector [41] for data reduction.

Implementation Details. In most experiments, we adopt SimCLR [43] as the self-supervised learning algorithm. SimCLR [43] define the loss function for a positive pair of examples (i, j) as

$$\mathcal{L}_{SimCLR}(i,j) = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}[k \neq i] \exp(\text{sim}(z_i, z_j)/\tau)} \tag{3}$$

where $\mathbb{1}[k \neq i]$ is an indicator function evaluating to 1 iff $k \neq i$, τ is a temperature parameter, z represents the projection of an input after being processed by the encoder

TABLE IX: Results for model extraction attacks against watermarking schemes on ImageNet dataset, where the scores for the best performance are bolded.

Watermarking Methods	Victim Models		Benign Models	Surrogate Models					
	Acc. (%)	WSR (%)	WSR (%)	Attack methods	Acc. (%)	WSR (%)	p-value$_w$	p-value$_b$	r
Margin-based [25]	70.06	100.00	4.62 ± 3.05	Retraining [29]	55.68	31.12	10^{-3}	10^{-53}	10^{49}
				Knockoff Nets [30]	63.24	47.35	10^{-14}	10^{-9}	10^{-6}
				AugSteal [36]	55.65	37.15	10^{-11}	10^{-18}	10^{7}
				D-DAE [69]	67.92	34.51	10^{-10}	10^{-19}	10^{9}
				MEBooster [70]	69.10	42.50	10^{-20}	10^{-19}	10^{-1}
				SSLExtraction (Ours)	68.37	**5.37**	$\mathbf{10^{-3}}$	$\mathbf{10^{-73}}$	$\mathbf{10^{70}}$
MAT [27]	74.25	100.00	35.29 ± 3.35	Retraining [29]	65.36	40.56	10^{-8}	10^{-83}	10^{74}
				Knockoff Nets [30]	72.51	56.17	10^{-26}	10^{-56}	10^{29}
				AugSteal [36]	70.10	59.51	10^{-31}	10^{-42}	10^{10}
				D-DAE [69]	71.98	66.87	10^{-35}	10^{-41}	10^{6}
				MEBooster [70]	70.25	67.24	10^{-35}	10^{-43}	10^{7}
				SSLExtraction (Ours)	70.97	**34.08**	$\mathbf{10^{-5}}$	$\mathbf{10^{-118}}$	$\mathbf{10^{113}}$
EWE [21]	73.91	96.50	2.87 ± 1.33	Retraining [29]	72.77	38.14	10^{-17}	10^{-15}	10^{-2}
				Knockoff Nets [30]	71.82	43.69	10^{-20}	10^{-11}	10^{-9}
				AugSteal [36]	70.52	38.90	10^{-17}	10^{-13}	10^{-4}
				D-DAE [69]	70.63	41.72	10^{-35}	10^{-13}	10^{-23}
				MEBooster [70]	70.82	45.30	10^{-35}	10^{-35}	10^{-35}
				SSLExtraction (Ours)	70.61	**3.96**	$\mathbf{10^{-1}}$	$\mathbf{10^{-72}}$	$\mathbf{10^{70}}$
MEA-Defender [71]	70.29	97.99	3.86 ± 2.58	Retraining [29]	64.26	35.97	10^{-9}	10^{-43}	10^{33}
				Knockoff Nets [30]	64.98	27.19	10^{-7}	10^{-37}	10^{30}
				AugSteal [36]	63.74	16.60	10^{-4}	10^{-50}	10^{46}
				D-DAE [69]	69.02	39.84	10^{-28}	10^{-20}	10^{-8}
				MEBooster [70]	69.13	42.30	10^{-34}	10^{-15}	10^{-19}
				SSLExtraction (Ours)	69.79	**5.84**	$\mathbf{10^{-1}}$	$\mathbf{10^{-93}}$	$\mathbf{10^{-91}}$

TABLE X: Results for our method against watermarking schemes on MNIST dataset.

Watermarking Methods	Victim Models		Benign Models	Surrogate Models					
	Acc. (%)	WSR (%)	WSR (%)	Attack methods	Acc. (%)	WSR (%)	p-value$_w$	p-value$_b$	r
Margin-based [25]	98.42	100.00	0.38 ± 0.71	SSLExtraction (Ours)	98.35	4.12	10^{-1}	10^{-92}	10^{91}
MAT [27]	98.31	100.00	32.74 ± 1.89	SSLExtraction (Ours)	98.22	30.18	10^{-18}	10^{-102}	10^{96}
EWE [21]	97.84	28.11	0.41 ± 1.02	SSLExtraction (Ours)	98.44	4.93	10^{-1}	10^{-93}	10^{91}
MEA-Defender [71]	97.92	100.00	0.85 ± 0.97	SSLExtraction (Ours)	98.06	2.95	10^{-1}	10^{-98}	10^{97}

and the projection head and $\mathrm{sim}(u,v) = \frac{u^\top v}{\|u\|\|v\|}$ is the cosine similarity. By sampling a large batch of inputs, the contrastive loss is applied to pairs of inputs within the batch. SimCLR [43] treats the other $N-1$ inputs in the batch as negative samples, allowing the model to learn by maximizing the similarity of positive pairs and minimizing the similarity of negative pairs.

To adapt ResNet-50 for our method, we make some modifications to enhance the model's ability to learn intrinsic feature representations without labeled data. Specifically, we remove the final fully connected classification layer and use a projection head, a two-layer MLP that maps the extracted features to a lower-dimensional space suitable for contrastive learning. Additionally, we apply extensive data augmentation techniques and optimize the model using the contrastive loss defined in Equation (3). To reduce training time, we train the model for 100 epochs and set the batch size to 256 in our experiments, which is significantly smaller than the recommended 4096 [43]. In our main experiments, we primarily adopt the SimCLR [43] framework. Additionally, we conduct experiments with MoCo v2 [44] and BYOL [45], along with an analysis of the impact of different training epochs and batch sizes, which are presented in Section IV-E.

TABLE XI: Results of our model extraction attack against different defensive methods on CIFAR-10.

Defense Methods	Victim Acc. (%)	Surrogate Acc. (%)
MODELGUARD [85]	93.70	92.41
QUEEN [86]	90.01	88.54
SNE [70]	88.23	90.57

To evaluate the effectiveness of watermark removal and data reduction, we follow the widely used linear evaluation protocol [43], [103], [104], where a linear classifier is trained on top of the frozen network obtained through contrastive learning. This linear layer is optimized using queries to the victim model and the corresponding hard-label outputs, enabling the surrogate model to extract knowledge from the victim model while maintaining the learned feature representations.

Appendix B
Proof of NP-Completeness for Problem (2)

(Theorem III.1). Optimization Problem (2) is NP-complete.

TABLE XII: Results for extracting the margin-based watermarked model under different victim-class absence scenarios.

Scenario	Victim Models		Surrogate Models	
	Acc. (%)	WSR (%)	Acc. (%)	WSR (%)
CIFAR-10: 1-Class Missing	87.81	100.00	87.16	3.94
CIFAR-10: 2-Class Missing	87.81	100.00	86.42	5.89
CIFAR-10: 5-Class Missing	87.81	100.00	83.15	7.03
CIFAR-10: Dog-Class Missing	87.81	100.00	87.19	4.16
ImageNet: Dog-Family Missing	70.06	100.00	66.91	6.41

Proof. First, we need to prove that the Problem (2) **belongs to NP**. Given a candidate solution b_i, we can verify its feasibility in polynomial time. We compute $\sum_{i=1}^{n} b_i$ and check whether it equals p, which takes $O(n)$ time. Then, the objective function in Problem (2) requires $O(n^2)$ time to compute. This shows that the problem belongs to NP.

Then we prove that the Problem (2) is **NP-hard**, we will reduce the Maximum Independent Set problem, which is known to be NP-hard [105], to our problem in polynomial time. The Maximum Independent Set Problem is a classical problem in graph theory. Given an undirected graph $G = (V, E)$ and $p < |V|$, the goal is to find a subset of vertices $S \subset V$ such that no two vertices in S are adjacent, and $|S| \geq p$.

We reduce the Maximum Independent Set problem to our problem by mapping the graph $G = (V, E)$ to a set of high-dimensional features. For each vertex $v_i \in V$, we define a corresponding feature h_i in high-dimensional space. The distance q_{ij} between any two features h_i and h_j is defined based on the adjacency in the graph:

$$q_{ij} = \mathbb{1}\left[(v_i, v_j) \notin E\right].$$

The goal of the Maximum Independent Set problem is to select a set of vertices where no two vertices are adjacent. In Problem (2), we are selecting a subset of features such that the sum of the distances between the selected features is maximized. When a graph G contains an independent set of size at least p, there exists a solution to Problem (2) that selects a subset H with pairwise distances equal to 1, where $|H| \geq p$. Since h_i and v_i have a one-to-one correspondence, the resulting set H of selected features in Problem (2) thus directly corresponds to the desired independent set in G.

Since the Maximum Independent Set problem is NP-hard, and we reduce it to our problem in polynomial time, this implies that our problem is NP-hard as well. Therefore, we have proven that Problem (2) is NP-complete. □

Appendix C
Approximation Ratio for Algorithm 1

To evaluate the theoretical performance of Algorithm 1 for solving the p-dispersion-sum problem defined in Equation (2), we analyze its approximation ratio, which quantifies how close the objective value obtained by the algorithm is to the optimal solution. Since the problem is NP-hard due to Theorem III.1, obtaining an exact solution in polynomial time is intractable. Therefore, approximation analysis provides a meaningful performance guarantee. In this section, we derive a lower bound on the ratio between the value returned by Algorithm 1 and the optimal objective value.

Theorem C.1. *Let I be an instance of the optimization problem given in Equation* (2). *Let* $\mathrm{OPT}(I)$ *and* $\mathrm{ALG}(I)$ *denote the objective values of the optimal solution and the solution returned by Algorithm 1 on instance I, respectively. Then the following approximation guarantee holds:*

$$\frac{\mathrm{OPT}(I)}{\mathrm{ALG}(I)} \leq 2. \tag{4}$$

Proof. For disjoint non-empty sets A, B, define $d(A, B) = \sum_{x \in A, y \in B} d(x, y)$ and $d(A) = d(A, A)$. We use two basic observations: (i) there exists $x \in A$ with $d(x, B) \geq d(A, B)/|A|$ (averaging); (ii) if $|B| \geq 2$, then $d(A, B) \geq |A|\, d(B)/(|B| - 1)$ (triangle-inequality based).

Let $\ell^* = \mathrm{OPT}(I)/\binom{k}{2}$. We prove by induction that

$$d(P_p) \geq \frac{p(p-1)}{2} \cdot \frac{\ell^*}{2}.$$

For $p = 2$, the algorithm selects the maximum-distance pair, which has distance at least ℓ^*.

Assume $|P_{k+1}| = k + 1$ satisfies the bound. We show that some $x \notin P_{k+1}$ satisfies

$$d(x, P_{k+1}) \geq \frac{k+1}{2}\,\ell^*.$$

Let P^* be an optimal set, $X = P^* \setminus P_{k+1}$ and $Y = P^* \cap P_{k+1}$.

Case 1: $X = P^*$. By (ii), $d(P_{k+1}, P^*) \geq (k+1)p\,\ell^*/2$. Applying (i) to $A = P^*$ yields $d(x, P_{k+1}) \geq \frac{k+1}{2}\,\ell^*$.

Case 2: $|X| \leq 1$. Then P_{k+1} contains all or all-but-one optimal elements; the next selected element contributes at least $(k+1)\ell^*/2$.

Case 3: $|X| \geq 2$. Since $d(P^*) = d(X) + d(Y) + d(X, Y)$, either $d(X) \geq \frac{1}{2} d(P^*)$ or $d(Y) + d(X, Y) \geq \frac{1}{2} d(P^*)$.

If $d(X) \geq \frac{1}{2} d(P^*)$, then (ii) on (P_{k+1}, X) gives

$$d(P_{k+1}, X) \geq (k+1) d(P^*)/(2(|X| - 1)).$$

Using (i) on $A = X$ yields $d(x, P_{k+1}) \geq \frac{k+1}{2}\,\ell^*$.

If $d(Y) + d(X, Y) \geq \frac{1}{2} d(P^*)$ and $|Y| = 1$, then $d(X, P_{k+1}) \geq d(X, Y) \geq \frac{1}{2} d(P^*)$, and (i) gives $d(x, P_{k+1}) \geq \frac{k+1}{2}\,\ell^*$. If $|Y| \geq 2$, applying (ii) to $(Y, P_{k+1} \setminus Y)$ gives a lower bound on $d(P_{k+1})$, and applying (ii) again to (X, P_{k+1}) yields

$$d(X, P_{k+1}) \geq |X| p\,\ell^*/2,$$

from which (i) implies $d(x, P_{k+1}) \geq \frac{k+1}{2}\,\ell^*$.

Since the algorithm picks the maximizer of $d(x, P_{k+1})$,

$$d(x, P_{k+1}) \geq \frac{k+1}{2}\,\ell^*$$

gives

$$d(P_{k+2}) \geq d(P_{k+1}) + (k+1)\frac{\ell^*}{2} = \frac{(k+1)(k+2)}{2} \cdot \frac{\ell^*}{2},$$

completing the proof.

□