DiffIP: Representation Fingerprints for Robust IP Protection of Diffusion Models

Zhuoling Li[1] Haoxuan Qu[1] Jason Kuen[2] Jiuxiang Gu[2]
Qiuhong Ke[3] Jun Liu[1*] Hossein Rahmani[1]
[1] Lancaster University [2] Adobe Research [3] Monash University
{z.li81, h.qu5}@lancaster.ac.uk, {kuen, jigu}@adobe.com,
Qiuhong.Ke@monash.edu, {j.liu81, h.rahmani}@lancaster.ac.uk

Abstract

Intellectual property (IP) protection for diffusion models is a critical concern, given the significant resources and time required for their development. To effectively safeguard the IP of diffusion models, a key step is enabling the comparison of unique identifiers (fingerprints) between suspect and victim models. However, performing robust and effective fingerprint comparisons among diffusion models remains an under-explored challenge, particularly for diffusion models that have already been released. To address this, in this work, we propose ***DiffIP****, a novel framework for robust and effective fingerprint comparison between suspect and victim diffusion models. Extensive experiments demonstrate the efficacy of our framework.*

1. Introduction

Recently, diffusion models have emerged as powerful tools, enabling the general public to easily generate high-quality images, such as cartoon illustrations [27], architectural design sketches [7], and advertising posters [69]. However, the development of diffusion models is typically resource- and time-intensive, leading advanced diffusion models (e.g., FLUX [2] and DeepFloyd [3]) to be released typically with specific licenses [1, 4] when open-sourced. These licenses protect the intellectual property (IP) of open-source diffusion models and often restrict their use within the community to non-commercial and non-production purposes. Unfortunately, violations of these licenses are common, with wrapped or fine-tuned versions of open-source diffusion models being misused to generate images in various styles for commercial gain [6]. Given this, it is essential for open-source diffusion model owners to identify whether a suspect model is a derivative of their original (victim) diffusion model or is independently developed. During such identification, especially in legal disputes, the general process typically involves owners of both the suspect and victim models presenting their respective models to an authoritative third party for arbitration [47, 61]. In this context, a critical step is enabling *the comparison of unique identifiers (fingerprints)* between the suspect and victim models to ascertain whether model theft or subsequent development has occurred.

Existing research on fingerprint comparison primarily fall into two categories: *external-watermark-injection* methods and *intrinsic-fingerprint-identification* methods. *External-watermark-injection* methods [19, 25, 46, 66] generally inject artificial fingerprints (i.e., watermarks) into the victim model, allowing for later extraction and identity verification. However, the process of injecting artificial watermarks can incur extra training costs and impair model performance [52]. More importantly, such a process is inapplicable to models that have already been released as open-source to the community. In light of these limitations, *intrinsic-fingerprint-identification* methods [16, 32, 50, 68, 71] have emerged as a promising alternative across various network types, including large language models and image classifiers. This line of methods considers using the model's inherent and native attributes as its *intrinsic fingerprint*, making them free of external watermark injection thus feasible for even already-released models [71]. Specifically, rather than directly using model weights [68, 74] as *intrinsic fingerprints*, a well-established observation is that a model's feature representation generally exhibits enhanced robustness compared to its weights during fine-tuning [18, 33, 58, 75]. Building on this insight, many recent *intrinsic-fingerprint-identification* methods [16, 32, 40, 43, 71] have focused on using a model's feature representations as its *intrinsic fingerprints*.

Given the advantages of representation-based *intrinsic-fingerprint-identification* methods, in this work, we aim to protect the IP of open-source diffusion models also from

*Corresponding Author

this perspective. However, despite the progress made in this line of works across various types of open-source models, using feature representation as *intrinsic fingerprints* for diffusion models' IP protection remains challenging due to two reasons. **(1)** We observe that, although the feature representations of diffusion models are generally more robust than their weights, such representations still commonly exhibit non-negligible changes during fine-tuning (see Fig. 1 in supplementary). This makes it still challenging to determine whether a suspect diffusion model has been fine-tuned from a stolen victim model by directly calculating the distance between their feature representations. **(2)** Existing representation-based *intrinsic-fingerprint-identification* methods primarily focus on protecting **deterministic networks** whose feature representations remain deterministic as long as the input is fixed. In contrast, the denoising process in diffusion models is often formulated as a stochastic differential equation [57], framing them as inherently **stochastic and temporal networks**. In other words, given an input z (e.g., a text prompt for text-to-image diffusion models), the feature representation of a diffusion model is inherently a **temporal sequence** $\{x_t\}_{t=1}^{T}$. In $\{x_t\}_{t=1}^{T}$, each step x_t is not deterministic but rather a sample drawn from a **stochastic probability distribution** $F(\cdot|z,t)$ conditioned on the input z and time step t [23, 41]. Such differences between deterministic networks and stochastic and temporal networks further hinder the efficacy of existing representation-based *intrinsic-fingerprint-identification* methods in diffusion models' IP protection.

In this paper, we propose **DiffIP**, a novel feature representation-based framework as the first *intrinsic-fingerprint-identification*-type IP protection method specifically designed for diffusion models. DiffIP enables effective and convenient IP protection of diffusion models, even for those already released ones. Specifically, to overcome the two challenges highlighted above, our DiffIP comprises two corresponding designs, as follows.

Firstly, to mitigate the impact of fine-tuning on feature representations and enhance the robustness of representation-based fingerprint comparison, a promising approach is to revert the feature representation of the suspect diffusion model (F_s) back to the state of the victim diffusion model (F_v) before comparing their fingerprints (feature representations). However, achieving such reversion is still challenging due to the need to model the fine-tuning process. Diffusion model fine-tuning is typically performed to adapt models to accommodate domain shifts of downstream tasks (e.g., new styles), and previous works [37, 38, 48, 49, 54, 64] have shown that the effect of such fine-tuning on the feature distribution shift can be approximated using a linear transformation comprising orthogonal, scaling, and translation operators. Inspired by these insights, we design a novel method within our DiffIP framework to revert F_s's feature representation to F_v's state through a linear-approximation-based pipeline, enabling robust fingerprint comparison between F_v and F_s.

Moreover, recognizing the **stochastic** and **temporal** nature of diffusion models, we further formulate a novel metric to quantify the representation similarity between the victim diffusion model F_v and suspect diffusion model F_s. To facilitate the proper comparison between F_v and F_s, given an input z (e.g., a text prompt), we aim to compute the similarity between their corresponding **stochastic** and **temporal** distribution sequences $\{F_v(\cdot|z,t)\}_{t=1}^{T_v}$ and $\{F_s(\cdot|z,t)\}_{t=1}^{T_s}$. To achieve this, we first use diverse random seeds to trigger F_v and F_s and collect their responses, akin to performing multiple samples from **stochastic distributions** $F_v(\cdot|z,t)$ and $F_s(\cdot|z,t)$ to estimate their underlying properties. Moreover, we introduce a dynamic-programming-based mechanism to address potential misalignment in the distribution sequences of F_v and F_s, with special consideration into the **temporal nature** of diffusion models. The underlying idea is to find the best alignment between $\{F_v(\cdot|z,t)\}_{t=1}^{T_v}$ and $\{F_s(\cdot|z,t)\}_{t=1}^{T_s}$ through nonlinear matching, minimizing their total cumulative distance. Such minimal total distance can then be viewed as the final dissimilarity between F_v and F_s for a given input z.

With the above designs properly integrated, our DiffIP can effectively and efficiently determine if a suspect diffusion model originates from a victim diffusion model. Furthermore, as analyzed in Sec. 3.3, DiffIP remains effective even when the owner of the suspect model employs various camouflage techniques [68], including dimension permutation and column-wise scaling, to evade fingerprint verification. This further enhances DiffIP's practicality.

2. Related Work

Model IP Protection. With the increasing commercial value and high development costs of advanced deep learning models, model IP protection has become a critical concern for model developers, attracting significant attention from both academia and industry [42, 45, 55, 58, 71]. To achieve model IP protection, a promising approach is comparing unique identifiers (fingerprints) between suspect and victim models. Recent fingerprint comparison methods primarily fall into two main categories: *external-watermark-injection* methods [19, 20, 24, 25, 39, 62, 65, 70, 72] and *intrinsic-fingerprint-identification* methods [16, 32, 40, 43, 68, 71, 74]. Existing *external-watermark-injection* methods have explored embedding identifiers (watermarks) into various models [20, 70], including diffusion models [24, 25]. However, these approaches require modifying model parameters (e.g., through training), making them unsuitable for protecting already released models [71]. This limitation has driven growing interest in *intrinsic-fingerprint-identification* methods [68, 71, 74], which require no addi-

tional training and can be applied to released models.

In *intrinsic-fingerprint-identification* methods, researchers leverage inherent (intrinsic) attributes of well-trained models, rather than injected watermarks, as IP identifiers (fingerprints) for fingerprint comparison. Early approaches primarily explored model weights [68, 74] as model's *intrinsic fingerprints*. However, weight-based methods are generally highly sensitive to minor perturbations, limiting their robustness in practical scenarios [58]. In light of this, many recent works [16, 32, 71] have shifted towards using model's feature representations as *intrinsic fingerprints*. Among them, Zhang et al. [71] proposed to perform model IP protection via computing representation similarity between victim and suspect models using Centered Kernel Alignment (CKA) [35]. Meanwhile, Jia et al. [32] proposed to analyze the graphical differences of feature representations to protect graph neural networks.

In this paper, we also explore using feature representations as models' *intrinsic fingerprints* during fingerprint comparison and model IP protection. Different from previous *intrinsic-fingerprint-identification* methods, our work considers the distinct characteristics of diffusion models, designing the first intrinsic-fingerprint-based IP protection method specifically tailored to diffusion models.

Diffusion Models [22, 29, 51, 56] have emerged as powerful tools for image generation, finding applications in lots of areas such as cartoon image creation [27] and medical image synthesis [15]. Despite their impressive capabilities, the IP protection of (released) diffusion models remains an under-explored challenge. In this work, we introduce DiffIP, a novel framework that enables robust IP protection for diffusion models based on *intrinsic fingerprint* comparison.

3. Proposed Method: DiffIP

In this paper, to support IP protection of diffusion models, the specific goal is to determine whether a suspect diffusion model (F_s) is derived (fine-tuned) from a victim diffusion model (F_v). To achieve this, the core idea behind our method is to identify an optimal transformation that approximately reverts the feature representations of F_s back to the state of F_v. This transformation enables state alignment between the feature representations of F_v and F_s, aiding in an effective and robust fingerprint comparison between them.

A key step in performing such a fingerprint comparison is first determining the *intrinsic fingerprints* of diffusion models. We observe that, given an input z (e.g., a text prompt), the output of a diffusion model F throughout its denoising process forms a temporal sequence $\{F(\cdot|z,t)\}_{t=1}^{T}$, where each step $F(\cdot|z,t)$ is a stochastic distribution conditioned on z and time step t. Since this sequence captures both the stochastic and temporal aspects of the diffusion model—effectively reflecting its behavior—we use it as the diffusion model's *intrinsic fingerprint* in our framework. To ease understanding, in the rest of this work, we call $\{F(\cdot|z,t)\}_{t=1}^{T}$ a **fingerprint sequence**.

With diffusion model's *intrinsic fingerprints* determined above, let $\{F_s(\cdot|z,t_s)\}_{t_s=1}^{T_s}$ denote the fingerprint sequence of the suspect diffusion model F_s, and $\{F_v(\cdot|z,t_v)\}_{t_v=1}^{T_v}$ denote the fingerprint sequence of the victim diffusion model F_v. In our framework, the next phase is to revert $\{F_s(\cdot|z,t_s)\}_{t_s=1}^{T_s}$ to the same state as $\{F_v(\cdot|z,t_v)\}_{t_v=1}^{T_v}$, and then compare them at this state. However, this operation on multi-step fingerprint sequences is challenging. To address this, DiffIP decomposes the process into two sub-phases. In sub-phase 1 (introduced in Sec. 3.1), we first revert each step in $\{F_s(\cdot|z,t_s)\}_{t_s=1}^{T_s}$ individually and compare it with each step in $\{F_v(\cdot|z,t_v)\}_{t_v=1}^{T_v}$. Then, in sub-phase 2 (discussed in Sec. 3.2), we build on the results from sub-phase 1, account for the multi-step nature of fingerprint sequences, and effectively compare the entire suspect fingerprint sequence with the entire victim fingerprint sequence.

3.1. Sub-Phase 1: Comparison Between Steps

In this section, we discuss how our DiffIP compares the (dis)similarity between each step $F_s(\cdot|z,t_s)$ of the suspect fingerprint sequence $\{F_s(\cdot|z,t_s)\}_{t_s=1}^{T_s}$ and each step $F_v(\cdot|z,t_v)$ of the victim fingerprint sequence $\{F_v(\cdot|z,t_v)\}_{t_v=1}^{T_v}$. Here, $F_s(\cdot|z,t_s)$ represents the suspect diffusion model's feature distribution at time step t_s, while $F_v(\cdot|z,t_v)$ represents the victim diffusion model's feature distribution at time step t_v. To effectively compare $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$, an essential point is to first revert $F_s(\cdot|z,t_s)$ to the state of $F_v(\cdot|z,t_v)$, thereby mitigating the interference caused by fine-tuning in their comparison. Previous studies [37, 38, 48, 49, 54, 60, 64] have shown that the effects of fine-tuning on model feature distribution shift, can be approximated using a linear transformation comprising orthogonal, scaling, and translation operators. Building on this insight, we interpret the problem of effective similarity comparison between $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ as a two-step process: (1) applying an appropriate linear transformation (an appropriate composition of orthogonal, scaling, and translation) to revert $F_s(\cdot|z,t_s)$ to the state of $F_v(\cdot|z,t_v)$, and (2) computing the dissimilarity (distance) between $F_v(\cdot|z,t_v)$ and the transformed (reverted) version of $F_s(\cdot|z,t_s)$ at this same state (i.e., the state of $F_v(\cdot|z,t_v)$). Below, we elaborate on how we approach this problem based on this two-step interpretation.

Conceptual problem definition. To address this problem, we first formally define it as follows. Let M denote a linear transformation comprising orthogonal, translation, and scaling operators. According to its above interpretation, the problem can be formulated as an optimization problem, where the goal is to find an optimal M that, when applied to $F_s(\cdot|z,t_s)$, minimizes its dissimilarity to $F_v(\cdot|z,t_v)$. Specifically, taking this minimal dissimilarity

as the step-wise distance $d_{step}(\cdot,\cdot)$ between $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$, $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ can then be expressed as:

$$d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v)) := \min_M \mathbb{E}[dis(M(X),Y)] \quad (1)$$

where $X \sim F_s(\cdot|z,t_s)$, $Y \sim F_v(\cdot|z,t_v)$, and $dis(\cdot,\cdot)$ represents a distance function. Besides, for M as a linear transformation comprising orthogonal, scaling, and translation operators, without loss of generalizability, we can express M as $M(X) = \mathbf{QS}X - \mathbf{h}$, where $\mathbf{Q}$ is an orthogonal transformation matrix, $\mathbf{S}$ is a diagonal scaling matrix, and $\mathbf{h}$ is a translation vector.

Notably, according to the above two-step interpretation of the problem, $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ defined in this way can quantify the dissimilarity between $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ at the same state. In other words, a large $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ implies that $F_s(\cdot|z,t_s)$ remains irrelevant to $F_v(\cdot|z,t_v)$ after mitigating the interference of fine-tuning, whereas a small $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ indicates their relevance. Based on this understanding of $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$, the problem of performing effective similarity comparison between $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ then reduces to solving out $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$.

Practical problem formulation. However, the lack of explicit parametric representations for $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ as unknown stochastic distributions poses significant challenges in directly solving $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ from Eq. 1. To overcome this limitation and make $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ practically solvable, we aim to reformulate the optimization problem in Eq. 1 further into a form in which all terms, except M, are explicitly represented. To achieve this, since both $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ are stochastic distributions, we propose applying distribution sampling on them to estimate the behaviors of X and Y, as follows.

To estimate X's behavior for a given input z, we generate N diverse random seeds to trigger the suspect diffusion model F_s and collect N responses from it, where N is a hyperparameter. This process essentially draws N samples $(\{x_i\}_{i=1}^N)$ from $F_s(\cdot|z,t_s)$. Similarly, using the same N seeds to trigger the victim diffusion model F_v, we obtain $\{y_j\}_{j=1}^N$ where each $y_j \sim F_v(\cdot|z,t_v)$. With $\{x_i\}_{i=1}^N$ and $\{y_j\}_{j=1}^N$, we can estimate the behavior of X and Y based on the law of large numbers [9, 44]. Meanwhile, $\{x_i\}_{i=1}^N$ and $\{y_j\}_{j=1}^N$ as sets of samples now have explicit representations. In light of this, by leveraging $\{x_i\}_{i=1}^N$ and $\{y_j\}_{j=1}^N$ in replacement of X and Y, we can rewrite Eq. 1 as:

$$\begin{aligned} &d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v)) \\ &:= \min_M \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} dis(M(x_i),y_j), \end{aligned} \quad (2)$$

which now provides a practical way to solve $d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v))$ out.

Problem in Eq. 2 is still challenging. Despite its practical formulation, solving the optimization problem in Eq. 2 remains highly challenging due to the specially constrained search space for M. Specifically, as discussed above, M is composed of orthogonal, scaling, and translation operators. This constraint makes the naive application of common optimization strategies, such as gradient descent or least squares, inapplicable for optimizing M. In light of this, in the following parts, we then find a solution for this challenging problem.

Our exploration. To help readers better understand our thought process, which ultimately leads to our solving procedure of Eq. 2 (elaborated in the next part), we first present our exploration here. At the start of our exploration, we observe that, the optimization problem in Eq. 2 shares similarities with the Procrustes problem [5, 8, 53]. Specifically, the Procrustes problem, as a classical optimization problem that has a closed-form solution, is formulated as:

$$\min_{\mathbf{Q}} \|\mathbf{QA} - \mathbf{B}\|^2, \quad (3)$$

where the goal is to find an optimal orthogonal matrix $\mathbf{Q}$ that minimizes the L2 distance between two matrices $\mathbf{A}$ and $\mathbf{B}$. Similarly, by defining the distance function $dis(\cdot,\cdot)$ as the commonly used L2 distance, and based on the previously elaborated expression of M (i.e., $M(X) = \mathbf{QS}X - \mathbf{h}$), we can rewrite Eq. 2 as:

$$\begin{aligned} &d_{step}(F_s(\cdot|z,t_s),F_v(\cdot|z,t_v)) \\ =&\frac{1}{N^2}\Big(\min_{\{\mathbf{Q},\mathbf{S},\mathbf{h}\}} \|\mathbf{QSX_{sample}} - \mathbf{h} - \mathbf{Y_{sample}}\|^2\Big), \end{aligned} \quad (4)$$

where $\mathbf{X_{sample}}$ and $\mathbf{Y_{sample}}$ are two sample matrices respectively deduced from $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$. We provide the derivations of Eq. 4, as well as $\mathbf{X_{sample}}$ and $\mathbf{Y_{sample}}$, in supplementary. As shown, the optimization (minimal) problem in Eq. 4 is in a form similar to Eq. 3.

Based on the above, we then seek inspiration from Eq. 3 (already solvable as the Procrustes problem) to get clue into solving the optimization problem in Eq. 4. Specifically, we notice that, in Eq. 3, only the orthogonal matrix $\mathbf{Q}$ needs to be optimized, while in Eq. 4, the optimization problem requires optimizing $\{\mathbf{Q},\mathbf{S},\mathbf{h}\}$, which is simultaneously related to three terms: the orthogonal matrix $\mathbf{Q}$, the scaling matrix $\mathbf{S}$, and the translation vector $\mathbf{h}$. Building on this remaining key difference between the optimization problems in Eq. 3 and Eq. 4, we derive a procedure—elaborated below—that can be used to solve the optimization problem in Eq. 4, based on Theorem 1 and Theorem 2.

Theorem 1. *(Proof in Supplementary). The optimization problem in Eq. 4,* $\min_{\{\mathbf{Q},\mathbf{S},\mathbf{h}\}} \|\mathbf{QSX_{sample}} - \mathbf{h} - \mathbf{Y_{sample}}\|^2$,

can be rewritten into the following form with $\mathbf{h}$ *removed:*

$$\min_{\{\mathbf{Q},\mathbf{S}\}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2, \tag{5}$$

where $\mathbf{X}'_{\mathbf{sample}}$ *and* $\mathbf{Y}'_{\mathbf{sample}}$ *are two new matrices respectively derived from* $\mathbf{X}_{\mathbf{sample}}$ *and* $\mathbf{Y}_{\mathbf{sample}}$.

Theorem 2. *(Proof in Supplementary). The following optimization problem, when optimized over* $\mathbf{S}$*, has a practically derivable closed-form solution.*

$$\min_{\mathbf{S}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2. \tag{6}$$

How we solve the optimization problem in Eq. 4. Based on our exploration above, we now detail how we solve the optimization problem in Eq. 4. Specifically, leveraging Theorem 1, we first enable simplifying this problem to the one in Eq. 5. The remainder of this part then elaborates on how we solve the optimization problem in Eq. 5.

Inspired by the Alternating Least Squares (ALS) algorithm [36, 59], we observe that the optimization problem in Eq. 5, which involves optimizing over $\{\mathbf{Q}, \mathbf{S}\}$, is solvable if the following two sub-problems are independently solvable: (1) $\min_{\mathbf{Q}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$, which optimizes solely over $\mathbf{Q}$ and (2) $\min_{\mathbf{S}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$, which optimizes solely over $\mathbf{S}$. Meanwhile, based on our earlier exploration, we note that both these two sub-problems have closed-form solutions, allowing them to be solved efficiently. Specifically, $\min_{\mathbf{Q}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$ admits a closed-form solution as it is just a classical Procrustes problem in Eq. 3, with $\mathbf{A} = \mathbf{SX}'_{\mathbf{sample}}$ and $\mathbf{B} = \mathbf{Y}'_{\mathbf{sample}}$. At the same time, $\min_{\mathbf{S}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$ can also be solved in closed form using Theorem 2. Given the above, we can finally efficiently solve the optimization problem in Eq. 5 via alternatingly solving the two above sub-problems, similar to the ALS algorithm. The optimization terminates when the consecutive updates of both $\mathbf{Q}$ and $\mathbf{S}$ fall below a small threshold ϵ, where ϵ is a hyperparameter. A pseudo-code description of this alternating process is also provided in supplementary.

Final derivation of $d_{step}(F_s(\cdot|z,t_s), F_v(\cdot|z,t_v))$**.** Denote $\mathbf{Q}^*$ and $\mathbf{S}^*$ as the orthogonal matrix and the scaling matrix, respectively, obtained via the above process. We can finally compute the step-wise distance $d_{step}(\cdot,\cdot)$ between $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ as:

$$\begin{aligned} &d_{step}(F_s(\cdot|z,t_s), F_v(\cdot|z,t_v)) \\ =&\frac{1}{N^2} \times \left(\left\|\mathbf{Q}^*\mathbf{S}^*\mathbf{X}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2 \right). \end{aligned} \tag{7}$$

3.2. Sub-Phase 2: Comparison Between Fingerprint Sequences

Above, we discuss how DiffIP effectively compares individual steps (i.e., $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$) from the fingerprint sequences of suspect and victim diffusion models. Building on the step-wise comparison results, we here introduce how DiffIP extends to comparing entire fingerprint sequences, $\{F_s(\cdot|z,t_s)\}_{t_s=1}^{T_s}$ and $\{F_v(\cdot|z,t_v)\}_{t_v=1}^{T_v}$. For brevity, we henceforth abbreviate these two sequences as $F^s_{1:T_s}$ and $F^v_{1:T_v}$. Notably, despite having step-wise comparison results, effectively comparing entire fingerprint sequences remains non-trivial. Specifically, even when $F^s_{1:T_s}$ and $F^v_{1:T_v}$ have the same sequence length (i.e., $T_s = T_v$) and F_s is indeed derived (fine-tuned) from F_v, the denoising states (progress) of F_s and F_v at the same time step can still be misaligned, due to various techniques employed during the fine-tuning of F_s [26, 63]. Consequently, even in the simplified case where $T_s = T_v$, a naive one-to-one comparison at identical time steps between $F^s_{1:T_s}$ and $F^v_{1:T_v}$ may lead to misalignment. This issue becomes more pronounced in the general case where $T_s \neq T_v$, in which even the suboptimal one-to-one comparison is infeasible.

Given the above, to effectively compare $F^s_{1:T_s}$ and $F^v_{1:T_v}$, we need to establish a step-alignment plan that properly aligns steps across the two sequences that correspond to similar stages in the denoising process. To achieve this, a naive approach is to match each step in $F^s_{1:T_s}$ with its closest step in $F^v_{1:T_v}$, using $d_{step}(\cdot,\cdot)$ in Eq. 7 as the step-wise distance. This approach is based on the intuition that if two steps from different sequences exhibit a small distance, they likely correspond to similar stages in the denoising process and should be aligned. Yet, this naive method may violate the temporal constraint: The alignments in the plan need to be order-preserving, ensuring that the relative order of steps remains consistent with the original temporal sequences. Specifically, if step $F_s(\cdot|z,t_s)$ in $F^s_{1:T_s}$ is aligned with step $F_v(\cdot|z,t_v)$ in $F^v_{1:T_v}$, then $F_s(\cdot|z,t_s+1)$ cannot be aligned with any step in $F^v_{1:T_v}$ that appears before $F_v(\cdot|z,t_v)$.

Recognizing this problem, we incorporate the classical dynamic programming algorithm [10, 11, 21] into our DiffIP, which is typically is typically formulated as a recursive approach to solve optimal sequential decision problems . This recursive property of dynamic programming is highly compatible with our case, where we need to make "step-alignment" decisions over time. Next, we elaborate on the introduced dynamic programming algorithm. To ease understanding, we also illustrate the algorithm in Fig. 1.

As illustrated, we first construct a grid matrix $\mathbf{D} \in \mathbb{R}^{T_s \times T_v}$, where each element is defined as $\mathbf{D}(t_s,t_v) = d_{step}(F_s(\cdot|z,t_s), F_v(\cdot|z,t_v))$ (computed via Eq. 7). Our goal can then be interpreted as finding an optimal "path" that starts from $\mathbf{D}(1,1)$, traverses through the matrix $\mathbf{D}$, and reaches $\mathbf{D}(T_s,T_v)$ while minimizing the summation (total)

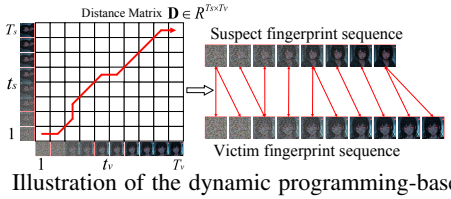

Figure 1. Illustration of the dynamic programming-based fingerprint sequence comparison, where $T_s = 8$ and $T_v = 10$. To compare two potentially mismatched fingerprint sequences, we first construct a cost matrix $\mathbf{C} \in \mathbb{R}^{T_s \times T_v}$ where each element $\mathbf{C}(t_s, t_v)$ represents the minimal total step-wise distance between $F_{1:t_s}$ and $F_{1:t_v}$. We update $\mathbf{C}$ recursively using a dynamic programming algorithm until $\mathbf{C}(T_s, T_v)$ is obtained, which corresponds to the minimal total step-wise distance between the two fingerprint sequences.

step-wise distance of the elements of $\mathbf{D}$ along this path. *Notably, when the path passes through a matrix grid $\mathbf{D}(t_s, t_v)$, it signifies that $F_s(\cdot|z, t_s)$ is aligned with $F_v(\cdot|z, t_v)$ in our step-alignment plan*. To achieve the above goal, following the classical dynamic programming strategy, we construct a total distance matrix $\mathbf{C} \in \mathbb{R}^{T_s \times T_v}$, where $\mathbf{C}(t_s, t_v)$ represents the minimal total distance between the subsequences $F^s_{1:t_s}$ and $F^v_{1:t_v}$. Naturally, we initialize $\mathbf{C}(1, 1) = \mathbf{D}(1, 1)$. Our objective is then to compute $\mathbf{C}(T_s, T_v)$, the minimal total distance between the full sequences $F^s_{1:T_s}$ and $F^v_{1:T_v}$. Before diving into this, let us consider a more general case: computing $\mathbf{C}(t_s, t_v)$. Recall that the formulated step-alignment plan need satisfy the constraint discussed earlier. Thus, the path traversing through the matrix $\mathbf{D}$ can only proceed to the right, upward, or right-up, as shown in Fig. 1. In other words, when the path reaches $\mathbf{D}(t_s, t_v)$, it can only come from its left, downward, or left-down directions. This insight lead to the following recursive formula for $\mathbf{C}(t_s, t_v)$:

$$\begin{aligned}&\mathbf{C}(t_s, t_v) = \mathbf{D}(t_s, t_v) + \\ &\min\{\mathbf{C}(t_s - 1, t_v), \mathbf{C}(t_s, t_v - 1), \mathbf{C}(t_s - 1, t_v - 1)\}.\end{aligned} \tag{8}$$

As shown, Eq. 8 as a recursive relation then allows us to compute $\mathbf{C}(T_s, T_v)$ iteratively from $\mathbf{C}(1, 1)$. A detailed algorithm of this process is also provided in supplementary.

After obtaining $\mathbf{C}(T_s, T_v)$ as the minimal total distance between $F^s_{1:T_s}$ and $F^v_{1:T_v}$, we can then simply define the distance between the two fingerprint sequences as $d_{\text{seq}}(F^s_{1:T_s}, F^v_{1:T_v}) = \mathbf{C}(T_s, T_v)$. Moreover, following [17], we can further normalize the distance $d_{\text{seq}}(F^s_{1:T_s}, F^v_{1:T_v})$ to the range $[0, 1]$, obtaining a similarity score S_{seq} that provides a more intuitive measure of the two sequences:

$$S_{\text{seq}}(F^s_{1:T_s}, F^v_{1:T_v}) = e^{-\frac{d_{\text{seq}}(F^s_{1:T_s}, F^v_{1:T_v})}{\sqrt{T_s T_v}}} \tag{9}$$

Based on $S_{\text{seq}}(F^s_{1:T_s}, F^v_{1:T_v})$, we can finally perform effective and robust fingerprint comparison between the suspect diffusion model F_s and the victim diffusion model F_v.

3.3. Further Discussion about DiffIP

The primary goal of DiffIP is to effectively compare the *intrinsic fingerprints* of the suspect diffusion model ($F^s_{1:T_s}$) with those of the victim diffusion model ($F^v_{1:T_v}$), ensuring that even fine-tuning of the suspect model does not obscure their derivation relationship. In this section, beyond this primary objective, we explore some interesting properties of DiffIP that may enhance its robustness in IP protection against some recent model camouflage techniques.

Recently, model camouflage techniques, including dimension permutation and column-wise scaling [68, 71], have emerged, allowing for noticeable modifications to model weights and intermediate representations without altering their final outputs or architectures. Given the existence of these techniques, skilled attackers may exploit them to further manipulate the suspect diffusion model, enabling it to evade fingerprint verification. This introduces an additional challenge in developing a robust and practical fingerprint comparison method—it needs to also be resilient to such camouflage techniques. Below, we also analyze that our DiffIP possesses an additional beneficial property: robustness against camouflage-based attacks, including dimension permutation and column-wise scaling.

Dimension permutation can permute the neuron arrangement of the suspect diffusion model, causing significant differences in its *intrinsic fingerprints* (feature representations) and complicating fingerprint verification. To address this, one approach is to adaptively apply dimension permutation also to the victim diffusion model's feature representation when comparing it with the suspect model's one, thereby eliminating discrepancies caused by different neuron arrangements. Formally, let $\mathbf{P}$ represent an optimizable permutation matrix. This approach integrates $\mathbf{P}$ also in Eq. 5, minimizing $\left\|\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{P}\mathbf{Y}'_{\mathbf{sample}}\right\|^2$ instead of $\left\|\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$ during step-wise comparison between $F_s(\cdot|z, t_s)$ and $F_v(\cdot|z, t_v)$. However, we here highlight that our DiffIP, without requiring any modifications, is inherently already identical to this approach. Specifically, since a permutation matrix is an isometric transformation [12, 13, 34], minimizing $\left\|\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{P}\mathbf{Y}'_{\mathbf{sample}}\right\|^2$ is equivalent to minimizing $\left\|\mathbf{P}^{-1}\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$. Here, as $\mathbf{P}^{-1}\mathbf{Q}$ remains an orthogonal matrix, minimizing $\left\|\mathbf{P}^{-1}\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2$ then reduces to the original Eq. 5. This demonstrates that our DiffIP is inherently robust to dimension permutation.

Column-wise scaling. Besides dimension permutation, attackers may also exploit column-wise scaling to manipulate the representations of the suspect diffusion model. Accordingly, our optimization objective defined in Eq. 5 is modified to minimize $\left\|c\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{P}\mathbf{Y}'_{\mathbf{sample}}\right\|^2$ instead, where $c \in \mathbb{R}^+$ is a positive scalar. Here, it is ev-

ident that this scaling term c can be combined with the scaling transformation matrix $\mathbf{S}$. Based on this combination, since $c\mathbf{S}$ is simply another scaling transformation matrix, minimizing $\left\|c\mathbf{Q}\mathbf{S}\mathbf{X}'_{\mathbf{sample}} - \mathbf{P}\mathbf{Y}'_{\mathbf{sample}}\right\|^2$, which is equivalent to $\left\|\mathbf{Q}(c\mathbf{S})\mathbf{X}'_{\mathbf{sample}} - \mathbf{P}\mathbf{Y}'_{\mathbf{sample}}\right\|^2$ can also be reduced back to Eq. 5. This shows that our DiffIP is also inherently robust to column-wise scaling.

3.4. Overall Process

Below, we discuss the overall fingerprint comparison process of our DiffIP framework, given a suspect diffusion model and a victim diffusion model. Specifically, for a given input (text prompt), we run both the suspect and victim diffusion models with N different random seeds, collecting their outputs at each step of the denoising process. This produces a fingerprint sequence for each model, denoted as $F^s_{1:T_s}$ and $F^v_{1:T_v}$. We then compute the similarity between $F^s_{1:T_s}$ and $F^v_{1:T_v}$ using Eq. 9. In our experiments, following [71], we use 200 inputs (text prompts in our case) for fingerprint comparison and report the average similarity score among these 200 inputs as the final result.

4. Experiments

To verify the effectiveness of our framework, we compare DiffIP with both existing intrinsic-fingerprint-based methods (in Sec. 4.1) and existing external-watermark-based methods (in Sec. 4.2), including REEF [71], PCS [68], Stable Signature[25], and AquaLoRA[24].

4.1. Comparison with Intrinsic-Fingerprint-based Methods

Baseline Methods. As most previous intrinsic-fingerprint-based IP protection methods are difficult to apply to diffusion models due to their specific designs for classification logits [16], GNN structures [32], or linguistic patterns in generated texts [31], we compare our DiffIP with two more recent methods, namely PCS [68] and REEF [71], which are compatible for IP protection of diffusion models. In addition, to better assess the effectiveness of different main components in our framework, we also assess three following variants of our DiffIP. **(1) DiffIP (w/o reversion)**: we remove the representation reversion scheme, modify Eq. 2, and directly calculate $d_{step}(F_s(\cdot|z,k_s), F_v(\cdot|z,k_v))$ as $\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\|x_i - y_j\|^2$. **(2) DiffIP (single sampling only)**: we disregard the stochastic nature of diffusion models by performing only a single sampling per input instead of multiple samplings. **(3) DiffIP (w/o DP)**: we remove our dynamic programming scheme (Sec. 3.2), and uniformly align steps in $F^s_{1:T_s}$ and $F^v_{1:T_v}$ for fingerprint sequences comparison, i.e., we align $F_s(\cdot|z,t_s)$ and $F_v(\cdot|z,t_v)$ if $\frac{t_s}{T_s} \approx \frac{t_v}{T_v}$.

SD15 as the Victim Model	**Derived Models (Similarity Score ↑)**						
	Fine-tuning				**Permutation**	**Scaling**	**Pruning**
Methods	Earth	Dream	Floor	Any	SD-Perm	SD-Scale	SD-Prun
PCS [68]	0.0682	0.9638	0.1197	0.9245	0.0000	0.9999	0.1625
REEF [71]	0.6419	0.4261	0.8430	0.4880	1.0000	1.0000	0.7586
DiffIP (w/o reversion)	0.0489	0.0533	0.0971	0.0734	0.0014	0.0310	0.0129
DiffIP (single sampling only)	0.7882	0.4969	0.7908	0.7593	0.9999	0.9999	0.7303
DiffIP (w/o DP)	0.7924	0.5930	0.7641	0.7843	0.9999	0.9999	0.7791
DiffIP	0.9953	0.8065	0.9892	0.9613	0.9999	0.9999	0.9573
— —							
FLUX as the Victim Model	**Derived Models (Similarity Score ↑)**						
	Fine-tuning				**Permutation**	**Scaling**	**Pruning**
Methods	Aes	Octane	Alpha	Portrait	FLUX-Perm	FLUX-Scale	FLUX-Prun
PCS [68]	0.0230	0.0231	0.0208	0.9926	0.0000	0.9999	0.1049
REEF [71]	0.9109	0.9292	0.9179	0.8797	1.0000	1.0000	0.4504
DiffIP (w/o reversion)	0.0670	0.0216	0.0111	0.0444	0.0574	0.0370	0.0078
DiffIP (single sampling only)	0.7667	0.7903	0.7801	0.6295	0.9999	0.9999	0.6380
DiffIP (w/o DP)	0.7768	0.7581	0.7876	0.6778	0.9999	0.9999	0.5778
DiffIP	0.8994	0.9380	0.9896	0.8783	0.9999	0.9999	0.8356

Table 1. Similarity of various intrinsic-fingerprint-based methods applied to derived diffusion models. denotes similarity > 0.8, for 0.5∼0.8, and for < 0.5.

SD15 as the Victim Model	**Unrelated Models (Similarity Score ↓)**			
Methods	FLUX	AnimeMiX	Lightning	Floyd
PCS [68]	0.0001	0.0001	0.0470	0.0000
REEF [71]	0.2607	0.5268	0.5417	0.2332
DiffIP (w/o reversion)	0.0118	0.0610	0.0604	0.0366
DiffIP (single sampling only)	0.0025	0.0019	0.0610	0.0418
DiffIP (w/o DP)	0.0034	0.0280	0.0826	0.0483
DiffIP	0.0150	0.1010	0.1006	0.0991
— —				
FLUX as the Victim Model	**Unrelated Models (Similarity Score ↓)**			
Methods	Any	Dream	Lightning	Floyd
PCS [68]	0.0001	0.0001	0.0002	0.0000
REEF [71]	0.5194	0.5242	0.5211	0.3115
DiffIP (w/o reversion)	0.0119	0.0146	0.0137	0.0743
DiffIP (single sampling only)	0.0363	0.0304	0.0472	0.0215
DiffIP (w/o DP)	0.0549	0.0228	0.0331	0.0497
DiffIP	0.0377	0.1001	0.0937	0.0991

Table 2. Similarity of various intrinsic-fingerprint-based methods applied to unrelated diffusion models. denotes similarity $<$ 0.2, for 0.2∼0.5, and for > 0.5.

Victim and Suspect Models. We follow [71] to collect a set of models to evaluate whether our DiffIP can distinguish models derived from the victim model from unrelated models. In this subsection, we use two widely adopted large pre-trained diffusion models, namely Stable Diffusion version 1.5 [51] (SD15) and FLUX-dev.1 (FLUX) [2], as victim models sequentially. For each victim model, we select a range of suspect diffusion models for evaluation, including seven diffusion models derived from the victim models and four unrelated diffusion models. Details of these suspect models are provided in Supplementary.

Implementation Details. In our experiments, following [71], we use 200 prompts from PartiPrompts [67] for fingerprint comparison and report the average similarity score among these 200 inputs as the final result. For each prompt, we perform sampling with 20 different random seeds (i.e., $N = 20$), and compute the representation similarity between the victim and suspect models. More implementation details are provided in Supplementary.

Results. Tab. 1 and Tab. 2 present the similarity between the victim models (SD15 and FLUX) and a series of suspect (derived and unrelated) models. As shown, although the derived models are developed using different techniques, compared to existing intrinsic-fingerprint-based methods and variants of our framework, DiffIP achieves consistently high representation similarity (with an average similarity of 0.9464) between the victim diffusion models and all

Methods	Model Fidelity FID ↓	Robustness against to Fine-tuning (TPR ↑) w/o fine-tuning	40 steps	1000 steps	10000 steps
Stable Signature [73]	24.77	0.9930	0.7735	0.0471	0.0094
AquaLoRA [24]	24.88	0.9900	0.0491	0.0000	0.0000
DiffIP	24.26	0.9591	0.9327	0.9259	0.9186

Table 3. Comparison between our DiffIP and external-watermark-based methods. We control the FPR at 10^{-6} and evaluate the TPR. SD15 is the victim model.

these derived models. In addition, the similarity computed by DiffIP remains consistently low for unrelated diffusion models (with an average similarity of 0.0807). This demonstrates that DiffIP can effectively distinguish diffusion models derived from victim models from unrelated ones and can serve as a reliable and robust fingerprinting method for IP protection.

4.2. Comparison with External-Watermark-based Methods

In this subsection, we further compare DiffIP with recent external-watermark-based methods, including AquaLoRA [24] and Stable Signature [73], to further evaluate its effectiveness. AquaLoRA and Stable Signature protect diffusion models by retraining them to embed a predefined 48-bit secret message into their generated outputs. This 48-bit message can then be decoded by a corresponding trained watermark decoder to verify model identity.

Settings. (1) Evaluation Metrics. Since watermark-based methods involve retraining diffusion models, we follow [24, 25] and use FID [28] to evaluate model fidelity, i.e., whether the watermarking process affects the generation quality of the model. To evaluate IP protection effectiveness, we also follow [24, 25] and use true positive rate (TPR), controlling false positive rate at 10^{-6} as an evaluation metric. (2) Personalized Fine-tuning. To evaluate the robustness of IP protection against personalized fine-tuning, we apply LoRA fine-tuning [30] to the watermarked SD15 model using a Naruto-style dataset [14], which contains 1,200 images. More details are provided in supplementary.

Results. Tab. 3 presents a comparison of our DiffIP with watermark-based methods. As shown, DiffIP achieves the best model fidelity since it does not require re-training diffusion models. Regarding IP protection effectiveness, watermark-based methods exhibit performance degradation after 40 fine-tuning steps, and their protection efficacy nearly disappears after 1000 fine-tuning steps. In contrast, our DiffIP remains robust even after more extensive fine-tuning (e.g., 1,0000 steps), This attributes to the specially design in our DiffIP, which actively reverts suspect models to the state of the victim model to mitigate the interference caused by fine-tuning.

4.3. Ablation Studies

Below, we use SD15 as the victim model, select Dream and Any as its derived models, and select FLUX, Lightning as its unrelated models to conduct ablation experiments.

Impact of Representation Reverting. Here, we extend our ablation study about the reversion scheme in the main experiments and further compare our method to the following variants: 1) **w/o Q**, where we remove the orthogonal operator $\mathbf{Q}$, which reduces M to only a combination of scaling and translation transformations. 2) **w/o S**, where we remove the scaling operator $\mathbf{S}$ from M. 3) **w/o t**, where we remove the translation operator $\mathbf{t}$ from M. 4) **Any** M, where we relax the constraint on M to allow any linear transformation and employ least squares to find the optimal M^*. 5) **w/o reversion** (discussed in Sec. 4.1). As shown in Tab. 4, our method achieves better performance than all other variants, demonstrating the efficacy of using a linear transformation composed of scaling, orthogonal, and translation transformations to approximately revert the fine-tuning process.

Method	Derived Models ↑ Dream	Any	Unrelated Models ↓ FLUX	Lightning
w/o $\mathbf{Q}$	0.5064	0.5467	0.0029	0.0113
w/o $\mathbf{S}$	0.4179	0.4643	0.0130	0.0229
w/o $\mathbf{t}$	0.3625	0.4365	0.0105	0.0201
Any M	0.9353	0.9686	0.4701	0.7703
w/o reversion	0.0533	0.0734	0.0118	0.0604
Ours	0.8065	0.9613	0.0150	0.1006

Table 4. Evaluation on the linear reversion scheme.

Impact of the Dynamic Programming Scheme. Here, we extend our ablation study about the dynamic programming scheme in the main experiments and further compare our method to the following variants: 1) **DiffIP-minimal**, where we ignore the temporal constraints in step alignment, and directly align each step in $F^s_{1:T_s}$ with the step in $F^v_{1:T_v}$ that has the minimal step-wise distance for fingerprint sequence comparison. 2) **DiffIP-last-step**, where we disregard the temporal nature of diffusion models and remove our dynamic programming scheme (Sec. 3.2). 3) **DiffIP-one-to-one**, where we manually set the suspect and victim diffusion models to have the same denoising steps (i.e., 25 steps), and thus we can align steps one-to-one in $F^s_{1:T_s}$ and $F^v_{1:T_v}$ (where $T_s, T_v = 25$) for fingerprint comparison. 4) **w/o DP** (discussed in Sec. 4.1). As shown in Tab. 5, our method yields the best results, indicating the efficacy of our dynamic programming scheme.

Method	Derived Models ↑ Dream	Any	Unrelated Models ↓ FLUX	Lightning
DiffIP-minimal	0.9613	0.9762	0.9011	0.9647
DiffIP-last-step	0.5843	0.7184	0.0017	0.0684
DiffIP-one-to-one	0.7224	0.8451	0.0109	0.0744
w/o DP	0.5930	0.7843	0.0034	0.0826
Ours	0.8065	0.9613	0.0150	0.1006

Table 5. Evaluation on the dynamic programming scheme.

5. Conclusion

In this work, we propose DiffIP, a novel framework designed to facilitate robust IP protection of diffusion models. We introduce several designs in DiffIP to promote the IP protection of diffusion models, accounting for their stochastic and temporal nature while also minimizing the interfere caused by (personalized) fine-tuning on their fingerprint comparison. Extensive experiments demonstrate the efficacy of DiffIP, indicating its potential as a practical solution for IP protection in diffusion models.

Acknowledgements. This work was partially supported by gift funding from Adobe Research and Catalyst funding from Lancaster University.

References

[1] LICENSE.md · black-forest-labs/FLUX.1-dev at main — huggingface.co. https://huggingface.co/black-forest-labs/FLUX.1-dev/blob/main/LICENSE.md. 1

[2] black-forest-labs/FLUX.1-dev · Hugging Face — huggingface.co. https://huggingface.co/black-forest-labs/FLUX.1-dev, . 1, 7

[3] DeepFloyd/IF-II-L-v1.0 · Hugging Face — huggingface.co. https://huggingface.co/DeepFloyd/IF-II-L-v1.0, . 1

[4] Deepfloyd If License - a Hugging Face Space by DeepFloyd — huggingface.co. https://huggingface.co/spaces/DeepFloyd/deepfloyd-if-license, . 1

[5] Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 4

[6] Stealing models for quick gains. https://www.bilibili.com/video/av1050913121, 2024. 1

[7] Zaina M Albaghajati, Donia M Bettaieb, and Raif B Malek. Exploring text-to-image application in architectural design: Insights and implications. *Architecture, Structures and Construction*, 3(4):475–497, 2023. 1

[8] Lars-Erik Andersson and Tommy Elfving. A constrained procrustes problem. *SIAM Journal on Matrix Analysis and Applications*, 18(1):124–139, 1997. 4

[9] Donald WK Andrews. Laws of large numbers for dependent non-identically distributed random variables. *Econometric theory*, 4(3):458–467, 1988. 4

[10] Richard Bellman. Dynamic programming. *science*, 153 (3731):34–37, 1966. 5

[11] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Princeton university press, 2015. 5

[12] Joel Bosveld, Arif Mahmood, Du Q Huynh, and Lyle Noakes. Constrained metric learning by permutation inducing isometries. *IEEE Transactions on Image Processing*, 25 (1):92–103, 2015. 6

[13] Deng Cai, Xiaofei He, Jiawei Han, et al. Isometric projection. In *AAAI*, pages 528–533, 2007. 6

[14] Eole Cervenka. Naruto blip captions. https://huggingface.co/datasets/lambdalabs/naruto-blip-captions/, 2022. 8

[15] Pierre Chambon, Christian Bluethgen, Curtis P Langlotz, and Akshay Chaudhari. Adapting pretrained vision-language foundational models to medical imaging domains. *arXiv preprint arXiv:2210.04133*, 2022. 3

[16] Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. Copy, right? a testing framework for copyright protection of deep learning models. In *2022 IEEE symposium on security and privacy (SP)*, pages 824–841. IEEE, 2022. 1, 2, 3, 7

[17] Shihyen Chen, Bin Ma, and Kaizhong Zhang. On the similarity metric and the distance metric. *Theoretical Computer Science*, 410(24-25):2365–2376, 2009. 6

[18] Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model, 2023. *URL https://arxiv. org/abs/2306.05720*. 1

[19] Hai Ci, Yiren Song, Pei Yang, Jinheng Xie, and Mike Zheng Shou. Wmadapter: Adding watermark control to latent diffusion models. *arXiv preprint arXiv:2406.08337*, 2024. 1, 2

[20] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, pages 485–497, 2019. 2

[21] Eric V Denardo. *Dynamic programming: models and applications*. Courier Corporation, 2012. 5

[22] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[23] Lyndon R Duong, Jingyang Zhou, Josue Nassar, Jules Berman, Jeroen Olieslagers, and Alex H Williams. Representational dissimilarity metric spaces for stochastic neural networks. *arXiv preprint arXiv:2211.11665*, 2022. 2

[24] Weitao Feng, Wenbo Zhou, Jiyan He, Jie Zhang, Tianyi Wei, Guanlin Li, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. Aqualora: Toward white-box protection for customized stable diffusion models via watermark lora. In *International Conference on Machine Learning*, pages 13423–13444. PMLR, 2024. 2, 7, 8

[25] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023. 1, 2, 7, 8

[26] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2023. 5

[27] Feihong He, Gang Li, Lingyu Si, Leilei Yan, Shimeng Hou, Hongwei Dong, and Fanzhang Li. Cartoondiff: Training-free cartoon image generation with diffusion transformer models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3825–3829. IEEE, 2024. 1, 3

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 8

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[30] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 8

[31] Dmitri Iourovitski, Sanat Sharma, and Rakshak Talwar. Hide and seek: Fingerprinting large language models with evolutionary learning. *arXiv preprint arXiv:2408.02871*, 2024. 7

[32] Ju Jia, Renjie Li, Cong Wu, Siqi Ma, Lina Wang, and Robert H Deng. Sigfinger: A subtle and interactive gnn fingerprinting scheme via spatial structure inference perturbation. *IEEE Transactions on Dependable and Secure Computing*, 2025. 1, 2, 3, 7

[33] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021. 1

[34] Paul Joseph Kelly. *On isometric transformations*. PhD thesis, 1942. 6

[35] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 3

[36] Pieter M Kroonenberg and Jan De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980. 5

[37] Xiaotong Li, Zixuan Hu, Yixiao Ge, Ying Shan, and Ling-Yu Duan. Exploring model transferability through the lens of potential energy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5429–5438, 2023. 2, 3

[38] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 2, 3

[39] Haozhe Liu, Wentian Zhang, Bing Li, Bernard Ghanem, and Jürgen Schmidhuber. Lazy layers to make fine-tuned diffusion models more traceable. *arXiv preprint arXiv:2405.00466*, 2024. 2

[40] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *arXiv preprint arXiv:1912.00888*, 2019. 1, 2

[41] Amin Nejatbakhsh, Victor Geadah, Alex H Williams, and David Lipshutz. Comparing noisy neural population dynamics using optimal transport distances. *arXiv preprint arXiv:2412.14421*, 2024. 2

[42] Guangyu Nie, Changhoon Kim, Yezhou Yang, and Yi Ren. Attributing image generative models using latent fingerprints. In *International Conference on Machine Learning*, pages 26150–26165. PMLR, 2023. 2

[43] Xudong Pan, Mi Zhang, Yifan Lu, and Min Yang. Tafa: A task-agnostic fingerprinting algorithm for neural networks. In *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*, pages 542–562. Springer, 2021. 1, 2

[44] Shige Peng. Law of large numbers and central limit theorem under nonlinear expectations. *arXiv preprint math/0702358*, 2007. 4

[45] Sen Peng, Yufei Chen, Jie Xu, Zizhuo Chen, Cong Wang, and Xiaohua Jia. Intellectual property protection of dnn models. *World Wide Web*, 26(4):1877–1911, 2023. 2

[46] Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7653–7668, 2023. 1

[47] John VH Pierce and Pierre-Yves Gunter. *The Guide to IP Arbitration*. Law Business Research Limited, 2022. 1

[48] Ryan Po, Guandao Yang, Kfir Aberman, and Gordon Wetzstein. Orthogonal adaptation for modular customization of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7964–7973, 2024. 2, 3

[49] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023. 2, 3

[50] Yehonathan Refael, Adam Hakim, Lev Greenberg, Tal Aviv, Satya Lokam, Ben Fishman, and Shachar Seidman. Slip: Securing llms ip using weights decomposition. *arXiv preprint arXiv:2407.10886*, 2024. 1

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 3, 7

[52] Mark Russinovich and Ahmed Salem. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique. *arXiv preprint arXiv:2407.10887*, 2024. 1

[53] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 4

[54] Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *European Conference on Computer Vision*, pages 286–302. Springer, 2022. 2, 3

[55] Hae Jin Song, Mahyar Khayatkhoei, and Wael AbdAlmageed. Manifpt: Defining and analyzing fingerprints of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10791–10801, 2024. 2

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2020. 2

[58] Yuchen Sun, Tianpeng Liu, Panhe Hu, Qing Liao, Shaojing Fu, Nenghai Yu, Deke Guo, Yongxiang Liu, and Li Liu. Deep intellectual property protection: A survey. *arXiv preprint arXiv:2304.14613*, 2023. 1, 2, 3

[59] Gábor Takács and Domonkos Tikk. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 83–90, 2012. 5

[60] Roberto Vega and Russell Greiner. Domain-shift adaptation via linear transformations. *arXiv preprint arXiv:2201.05282*, 2022. 3

[61] Dário Moura Vicente. Arbitrability of intellectual property disputes: a comparative survey. *Arbitration International*, 31 (1):163–170, 2015. 1

[62] Zilan Wang, Junfeng Guo, Jiacheng Zhu, Yiming Li, Heng Huang, Muhao Chen, and Zhengzhong Tu. Sleepermark: Towards robust watermark against fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2412.04852*, 2024. 2

[63] Mengfei Xia, Yujun Shen, Changsong Lei, Yu Zhou, Deli Zhao, Ran Yi, Wenping Wang, and Yong-Jin Liu. Towards more accurate diffusion model acceleration with a timestep tuner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5736–5745, 2024. 5

[64] Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4230–4239, 2023. 2, 3

[65] Jing Xu, Stefanos Koffas, Oğuzhan Ersoy, and Stjepan Picek. Watermarking graph neural networks based on backdoor attacks. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 1179–1197. IEEE, 2023. 2

[66] Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3277–3306, 2024. 1

[67] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 7

[68] Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. *Advances in Neural Information Processing Systems*, 37: 126332–126362, 2024. 1, 2, 3, 6, 7

[69] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion models in generative ai: A survey. *arXiv preprint arXiv:2303.07909*, 2023. 1

[70] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 159–172, 2018. 2

[71] Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In *Proceedings of the 2025 International Conference on Learning Representations (ICLR)*, 2025. 1, 2, 3, 6, 7

[72] Xiangyu Zhao, Hanzhou Wu, and Xinpeng Zhang. Watermarking graph neural networks by random graphs. In *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6. IEEE, 2021. 2

[73] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023. 8

[74] Yue Zheng, Si Wang, and Chip-Hong Chang. A dnn fingerprint for non-repudiable model ownership identification and piracy detection. *IEEE Transactions on Information Forensics and Security*, 17:2977–2989, 2022. 1, 2, 3

[75] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023. 1