DiffIP: Representation Fingerprints for Robust IP Protection of Diffusion Models

Supplementary Material

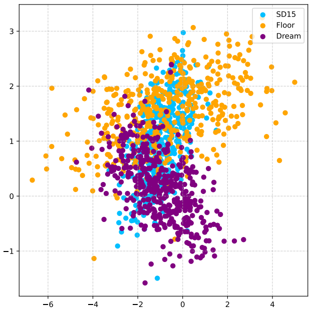

Figure 1. T-SNE visualization of feature distributions of diffusion models. Here, Floor and Dream are fine-tuned from SD15. As shown, compared to SD15, the feature distributions of its fine-tuned versions (i.e., Floor and Dream) exhibit shifts to varying degrees.

A. Proofs

A.1. Detailed Derivations of Eq.4 in the Main Paper

As discussed in Sec. 3.1, motivated by the classical Procrustes problem, we define the distance function $dis(\cdot,\cdot)$ in Eq. 2 of the main paper as the L2 distance. Thus, Eq. 2 can be rewritten as:

$$\min_{M} \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \|M(x_i) - y_j\|^2. \tag{1}$$

where $x_i, y_j \in \mathbb{R}^d$ are flattened feature representations. It can be observed that Eq. (1) represents a summation of the squared Euclidean norms of N^2 column vectors. Recall that the squared Frobenius norm of a matrix $\mathbf{A}$ is equivalent to the sum of the squared Euclidean norms of its column vectors, i.e.,

$$\|\mathbf{A}\|^2 = \sum_{i=1} \|a_i\|^2, \tag{2}$$

where a_i denotes the i-th column vector of matrix $\mathbf{A}$. With this concept in mind, we can naturally stack these N^2 column vectors into a matrix whose squared Frobenius norm is equal to the summation in Eq. (1). To achieve this, we first construct two sample matrices $\mathbf{X}_{\text{sample}}, \mathbf{Y}_{\text{sample}} \in \mathbb{R}^{d\times N^2}$ based on the summation structure in Eq. (1):

$$\begin{aligned}&\mathbf{X}_{\text{sample}} = \\ &[\underbrace{x_1, x_1, \dots, x_1}_{N}, \underbrace{x_2, x_2, \dots, x_2}_{N}, \dots, \underbrace{x_N, x_N, \dots, x_N}_{N}], \\ &\mathbf{Y}_{\text{sample}} = \\ &[\underbrace{y_1, y_2, \dots, y_N}_{N}, \underbrace{y_1, y_2, \dots, y_N}_{N}, \dots, \underbrace{y_1, y_2, \dots, y_N}_{N}],\end{aligned} \tag{3}$$

Then, we obtain the matrix $\mathbf{A} = M(\mathbf{X}_{\text{sample}}) - \mathbf{Y}_{\text{sample}}$. It is easy to observe that its column vectors correspond one-to-one with each term in the summation of Eq. (1). Consequently, we successfully transform Eq. 2 in the main paper into a form similar to the Procrustes problem, namely:

$$\min_{M} \frac{1}{N^2} \left\| M(\mathbf{X}_{\text{sample}}) - \mathbf{Y}_{\text{sample}} \right\|^2, \tag{4}$$

which can be further rewritten to Eq. 4 in the main paper by expanding M:

$$\frac{1}{N^2} \times \left(\min_{\{\mathbf{Q},\mathbf{S},\mathbf{h}\}} \left\| \mathbf{Q}\mathbf{S}\mathbf{X}_{\text{sample}} - \mathbf{h} - \mathbf{Y}_{\text{sample}} \right\|^2 \right). \tag{5}$$

A.2. Detailed Proof of Theorem 1

Here, we provide a detailed proof of Theorem 1. For ease of reading, we restate Theorem 1 from the main paper below.

Theorem 1. *The optimization problem in Eq. 4 of the main paper,* $\min_{\{\mathbf{Q},\mathbf{S},\mathbf{h}\}} \left\| \mathbf{QSX_{sample}} - \mathbf{h} - \mathbf{Y_{sample}} \right\|^2$*, can be rewritten into the following form with* $\mathbf{h}$ *removed:*

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

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

Proof. We begin by rewriting the norm term in Eq. 4 of the main paper as

$$\left\| \mathbf{B} - \mathbf{h} \right\|^2, \tag{7}$$

where $\mathbf{B} = \mathbf{QSX_{sample}} - \mathbf{Y_{sample}}$. Clearly, the optimal translation $\mathbf{h}^*$ that minimizes this norm term is given by $\mathbf{h}^* = \frac{\mathbf{B1}}{N^2}$ (N^2 is the number of column vectors of $\mathbf{B}$), which represents the row-wise mean of $\mathbf{B}$.

Applying this optimal translation to the elements of $\mathbf{B}$ is algebraically equivalent to removing the row means from $\mathbf{X_{sample}}$ and $\mathbf{Y_{sample}}$ separately. This operation is expressed as:

$$\begin{aligned}\mathbf{B} - \mathbf{h}^* =& \mathbf{B} - \frac{\mathbf{B1}}{N^2} = \mathbf{B}\left(\mathbf{I} - \frac{\mathbf{11}^\top}{N^2}\right) = \mathbf{BN} \\ =& \mathbf{QSX}_{\mathbf{sample}}\mathbf{N} - \mathbf{Y}_{\mathbf{sample}}\mathbf{N}\end{aligned} \tag{8}$$

where $\mathbf{N} = \mathbf{I} - \frac{\mathbf{11}^\top}{N^2}$ is the centering matrix. With this notation, we observe that $\mathbf{QS}(\mathbf{X}_{\mathbf{sample}}\mathbf{N}) = (\mathbf{QSX}_{\mathbf{sample}})\mathbf{N}$, which shows that the result is the same whether we center $\mathbf{X}_{\mathbf{sample}}$ before applying the transformation $\mathbf{QS}$ or center the transformed $\mathbf{QSX}_{\mathbf{sample}}$. Thus, we choose to center $\mathbf{X}_{\mathbf{sample}}$ and $\mathbf{Y}_{\mathbf{sample}}$ prior to transformation, eliminating the need to explicitly account for the translation operator $\mathbf{h}$. □

A.3. Detailed Proof of Theorem 2

Here, we provide a detailed proof of Theorem 2. For ease of reading, we restate Theorem 2 from the main paper below.

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

$$\min_{\mathbf{S}} \left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|^2. \tag{9}$$

Proof. We aim to minimize the objective function $\left\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\right\|$ with respect to the diagonal matrix $\mathbf{S}$, while keeping $\mathbf{Q}$ fixed. The Frobenius norm can be expanded as:

$$\begin{aligned}&\|\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}}\| = \\ &\mathrm{trace}\left((\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}})^\top(\mathbf{QSX}'_{\mathbf{sample}} - \mathbf{Y}'_{\mathbf{sample}})\right).\end{aligned} \tag{10}$$

Ignoring constant terms, the objective function simplifies to:

$$\begin{aligned}f(\mathbf{S}) = \mathrm{trace}(&\mathbf{S}^\top\mathbf{Q}^\top\mathbf{QSX}'_{\mathbf{sample}}\mathbf{X}'_{\mathbf{sample}}{}^\top - \\ &2\mathbf{Y}'_{\mathbf{sample}}{}^\top\mathbf{QSX}'_{\mathbf{sample}}).\end{aligned} \tag{11}$$

To find the optimal $\mathbf{S}$, we take the derivative of $f(\mathbf{S})$ with respect to the diagonal $\mathbf{S}$ and set it to zero. We then arrive at the matrix equation:

$$\begin{aligned}\mathrm{diag}&\left(\mathbf{Q}^\top\mathbf{QS}^*\mathbf{X}'_{\mathbf{sample}}\mathbf{X}'_{\mathbf{sample}}{}^\top\right) = \\ &\mathrm{diag}\left(\mathbf{X}'_{\mathbf{sample}}\mathbf{Y}'_{\mathbf{sample}}{}^\top\mathbf{Q}\right).\end{aligned} \tag{12}$$

Reminding that $\mathbf{S}$ is a diagonal matrix, we can rewrite the above equation to formulate a linear function as

$$\begin{aligned}\left[(\mathbf{Q}^\top\mathbf{Q}) \circ \left(\mathbf{X}'_{\mathbf{sample}}\mathbf{X}'_{\mathbf{sample}}{}^\top\right)\right]\mathbf{s}^* = \\ \mathrm{diag}(\mathbf{X}'_{\mathbf{sample}}\mathbf{Y}'_{\mathbf{sample}}{}^\top\mathbf{Q})\end{aligned} \tag{13}$$

where $\mathbf{s}^* = \mathrm{diag}(\mathbf{S}^*)$ is the vector form of the diagonal matrix $\mathbf{S}^*$. The solution s^* of the linear function follows directly from the inverse of the symmetric Hadamard product appearing on the left-hand side. □

B. Algorithms (Pseudo-Codes)

Alternating Algorithm to Solve the Optimization Problem in Eq. 4 of the Main Paper. We provide a pseudo-code of our alternating algorithm in Algorithm 1 to solve the optimization problem in Eq. 4 of the main paper, $\min_{\{\mathbf{Q},\mathbf{S},\mathbf{h}\}} \|\mathbf{QSX}_{\text{sample}} - \mathbf{h} - \mathbf{Y}_{\text{sample}}\|^2$.

Algorithm 1 Alternating Algorithm for Solving the Optimization Problem In Eq. 4 of the Main Paper

```
Require: Two sample matrices $\mathbf{X}_{\mathbf{sample}}$ and $\mathbf{Y}_{\mathbf{sample}}$,
    initial values for $\mathbf{Q}, \mathbf{S}$ (e.g., $\mathbf{Q}, \mathbf{S} = \mathbf{I}$), convergence
    threshold $\epsilon = 1 \times 10^{-4}$, and maximum iterations
    $T_{\text{max}} = 1000$
Ensure: Optimal values for $\mathbf{Q}^*$, $\mathbf{S}^*$
Row-wise center $\mathbf{X}_{\mathbf{sample}}$ and $\mathbf{Y}_{\mathbf{sample}}$ to obtain
    $\mathbf{X}'_{\mathbf{sample}}$ and $\mathbf{Y}'_{\mathbf{sample}}$. The optimization problem becomes:
        $\min_{\mathbf{Q},\mathbf{S}} \left\|\mathbf{QSX}'_{\text{sample}} - \mathbf{Y}'_{\text{sample}}\right\|^2$
Compute initial value $V_0 = \left\|\mathbf{QSX}'_{\text{sample}} - \mathbf{Y}'_{\text{sample}}\right\|^2$
Set iteration counter $i = 0$
while not converged and $i < T_{\text{max}}$ do
    Fix $\mathbf{S}$ and solve the minimization problem:
        $\min_{\mathbf{Q}} \left\|\mathbf{QSX}'_{\text{sample}} - \mathbf{Y}'_{\text{sample}}\right\|^2$
    Obtain $\mathbf{Q}^*$ using the closed-form solution in classical Procrustes problem
    Fix $\mathbf{Q}$ and solve the minimization problem:
        $\min_{\mathbf{S}} \left\|\mathbf{QSX}'_{\text{sample}} - \mathbf{Y}'_{\text{sample}}\right\|^2$
    Obtain $\mathbf{S}^*$ using the solution in Theorem 2
    Update $V = \left\|\mathbf{QSX}'_{\text{sample}} - \mathbf{Y}'_{\text{sample}}\right\|^2$
    if $V_0 - V < \epsilon$ then
        Exit the loop
    else
        Set $V_0 = V$
    end if
    Increment $i = i + 1$
end while
return $\mathbf{Q}^*, \mathbf{S}^*$
```

Dynamic Programming-based Fingerprint Comparison. We provide a pseudo-code of our dynamic programming algorithm in Algorithm 2.

Algorithm 2 Dynamic Programming-based Fingerprint Comparison

```
Require: Two fingerprint sequences $F_{1:T_s} = \{F(\cdot|z,t_s)\}_{t_s=1}^{T_s}$ and $F_{1:T_v} = \{F(\cdot|z,t_v)\}_{t_v=1}^{T_v}$
Ensure: The minimum total step-wise distance $\mathbf{C}(T_s, T_v)$ and the optimal step-wise alignment plan $P$
Initialize $\mathbf{C}$ as an $(T_s+1) \times (T_v+1)$ matrix with $\mathbf{C}(0, t_v) = \infty$ and $\mathbf{C}(t_s, 0) = \infty$ for all $t_s, t_v > 0$
Set $\mathbf{C}(0,0) = 0$
for $t_s = 1$ to $T_s$ do
    for $t_v = 1$ to $T_v$ do
        Compute local distance: $d(t_s, t_v) = d_{step}(F(\cdot|z,t_s), F(\cdot|z,t_v))$
        Update total distance: $\mathbf{C}(t_s,t_v) = d(t_s,t_v) + \min\{\mathbf{C}(t_s-1,t_v), \mathbf{C}(t_s,t_v-1), \mathbf{C}(t_s-1,t_v-1)\}$
    end for
end for
Backtracking:
Initialize empty path $P = []$
Set $(t_s, t_v) \leftarrow (T_s, T_v)$
while $t_s > 0$ and $t_v > 0$ do
    Append $(t_s, t_v)$ to $P$
    Find previous step: $(t'_s, t'_v) = \arg\min\{\mathbf{C}(t_s-1,t_v), \mathbf{C}(t_s,t_v-1), \mathbf{C}(t_s-1,t_v-1)\}$
    Update $(t_s, t_v) \leftarrow (t'_s, t'_v)$
end while
Reverse path $P$
return $\mathbf{C}(T_s, T_v), P$
```

C. More Details about Experiment Settings

C.1. More Implementation Details

As shown in Eq. 4 of the main paper, we use $N = 20$ different seeds to collect suspect and victim diffusion models' responses ($\{x_i\}_{i=1}^N$ and $\{y_j\}_{i=1}^N$) on each input text prompt to construct the sample matrices $\mathbf{X}_{\mathbf{sample}}$ and $\mathbf{Y}_{\mathbf{sample}}$ (details are provided in Appendix A.1). Here, x_i and y_j represent the *flattened* output features from the noise schedulers of the diffusion models. Noticing that the feature dimensionality of suspect and victim diffusion models may be different, we thus employ Principal Component Analysis (PCA) to map these flattened features into the same dimension $d = 20$. In addition, we set a threshold of $\epsilon = 10^{-4}$ to terminate the alternating algorithm (as outlined in Algorithm 1) and a maximum iteration limit of $T_{\max} = 1000$ to ensure the algorithm exits.

Moreover, to compare our DiffIP with external-watermark-based methods, we use DiffIP to compute the representation similarity between diffusion models on each input text prompt, and evaluate whether suspect diffusion models are derived from the protected diffusion model using a similarity threshold. The similarity threshold is determined by empirically controlling the false positive rate (FPR) below 10^{-6} following [9, 10], after which the true positive rate (TPR) is computed.

C.2. Details of Selected Suspect Models

Fine-tuning Models. Fine-tuning diffusion models is a common practice among both legitimate users and attackers. For SD15, we select its fine-tuned versions from Hugging Face [8], including EarthnDusk [7], DreamShaper [12], FloorPlanLoRA [15], and AnyLoRA [14] as suspect models, abbreviated as Earth, Dream, Floor, and Any, respectively. Similarly, for FLUX, we select its fine-tuned versions, including AestheticAnime [5], LoRA-Cinematic-Octane [1], Turbo-Alpha [2], and AWPortrait-FL [17], as suspect models, abbreviated as Aes, Octane, Alpha, and Portrait, respectively.

Pruned Models. Pruning is a widely used technique for model compression in edge applications [4] and can also serve as an effective method for intentional model camouflage. For SD15, we apply a recent structural pruning method [11] to obtain a variant model, denoted as SD15-Prun. For FLUX, we follow [18] to obtain its pruned version as a suspect model, abbreviated as FLUX-Prun.

Dimension Permutation and Scaling Transformation. Attackers may employ dimension permutation or column-wise scaling to substantially modify parameters of the victim model and for evading fingerprint detection. To evaluate the robustness of DiffIP, following [19, 20], we apply column-wise permutation and scaling to SD15 and FLUX, producing two variant models for each: SD15-perm and SD15-scale for SD15, and FLUX-perm and FLUX-scale for FLUX, which serve as their respective suspect models.

Unrelated Models. For SD15, we select a series of independently developed models as its unrelated models, including FLUX-dev1 [3], AAM_XL_AnimeMix [13], dreamshaper-xl-lightning [16], and DeepFloyd [6], abbreviated as FLUX, AnimeMix, Lightning, and Floyd. For FLUX, we use AnyLoRA [14], DreamShaper [12], dreamshaper-xl-lightning [16], and DeepFloyd [6], abbreviated as Any, Dream, Lightning, and Floyd.

D. More Ablation Studies

D.1. Impact of the Number of Sampling

In our main experiments, we use $N = 20$ different seeds to trigger the diffusion models for each input text prompt to perform sampling from the stochastic distribution. Here, we also explore the impact of using different sampling numbers N of neighbor frames for temporal attention computation. As shown

	Derived Models ↑		Unrelated Models ↓	
Method	Dream	Any	FLUX	Lightning
N=1	0.5287	0.7814	0.0074	0.0315
N=5	0.6124	0.8311	0.0048	0.0534
N=10	0.7789	0.9290	0.0119	0.0104
N=20	0.8065	0.9613	0.0150	0.1006
N=30	0.8064	0.9597	0.0132	0.1003

Table 1. Evaluation on the sampling number (N).

in Tab. 1, the performance improves noticeably when N is smaller than 20, and the improvement trend plateaus beyond this point. Based on this observation, we choose to set $N = 20$ in our experiments to achieve good results while maintaining efficiency.

D.2. Impact of the Representation Dimensions

Method	Derived Models ↑		Unrelated Models ↓	
	Dream	Any	FLUX	Lightning
d=10	0.7773	0.9476	0.0124	0.0596
d=15	0.7936	0.9598	0.0132	0.0750
d=20	0.8065	0.9613	0.0150	0.1006
d=30	0.8069	0.9620	0.0141	0.1004
d=40	0.8076	0.9627	0.0194	0.1026

Table 2. Evaluation on the representation dimension (d) .

In our main experiments, we use PCA as a pre-processing operation to map representation dimensionality of suspect and victim diffusion models to $d = 20$ for fingerprint comparison. Here, we also evaluate the impact of d. As shown in Tab. 2, the performance improves noticeably when d is smaller than 20, and the improvement tapers off later. We thus set $d = 20$ in our main experiments.

References

[1] aixonlab. Flux.1-dev-lora-cinematic-octane. https://huggingface.co/aixonlab/FLUX.1-dev-LoRA-Cinematic-Octane. 3

[2] alimama creative. Flux.1-turbo-alpha. https://huggingface.co/alimama-creative/FLUX.1-Turbo-Alpha. 3

[3] black-forest labs. Flux.1-dev. https://huggingface.co/black-forest-labs/FLUX.1-dev. 3

[4] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10850–10869, 2023. 3

[5] dataautogpt3. Flux-aestheticanime. https://huggingface.co/dataautogpt3/FLUX-AestheticAnime. 3

[6] DeepFloyd. If-i-xl-v1.0. https://huggingface.co/DeepFloyd/IF-I-XL-v1.0. 3

[7] EarthnDusk. Loras_2023. https://huggingface.co/EarthnDusk/Loras_2023. 3

[8] Hugging Face. Hugging face. https://huggingface.co/. 3

[9] 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. 3

[10] 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. 3

[11] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pages 381–399. Springer, 2024. 3

[12] Lykon. Dreamshaper. https://huggingface.co/Lykon/DreamShaper, . 3

[13] Lykon. Aam_xl_animemix. https://huggingface.co/Lykon/AAM_XL_AnimeMix, . 3

[14] Lykon. Anylora. https://huggingface.co/Lykon/AnyLoRA, . 3

[15] maria26. Floor_plan_lora. https://huggingface.co/maria26/Floor_Plan_LoRA. 3

[16] oguzm. dreamshaper-xl-lightning-dpmsde. https://huggingface.co/oguzm/dreamshaper-xl-lightning-dpmsde. 3

[17] Shakker-Labs. Awportrait-fl. https://huggingface.co/Shakker-Labs/AWPortrait-FL. 3

[18] TencentARC. Fluxkits. https://github.com/TencentARC/FluxKits. 3

[19] 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. 3

[20] 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. 3