LATENT COPYRIGHT SPACE FOR MODEL PROVENANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Model provenance and intellectual property protection are becoming increasingly important as trained models are treated as valuable digital assets and widely released, fine-tuned, and redistributed. A common approach is to embed ownership information into the model for subsequent verification. However, once a model is released and adapted to downstream tasks, the embedded ownership signals may be gradually weakened or even erased during fine-tuning. In this paper, we introduce a new concept, termed the *latent copyright space*, to capture such invariant characteristics of model derivation under downstream adaptation. Specifically, given a source model, we first construct a private low-dimensional space using a secret key. The source model is then minimally adjusted so that its projection into this space matches a prescribed copyright identity, with negligible effect on its original utility. After release, the model can be adapted using standard downstream training procedures. To verify a target model, we project it into the same copyright space and measure its consistency with the protected source. Models derived from the protected source are expected to preserve the encoded identity under downstream adaptation, whereas non-derived models exhibit lower identity consistency. Experimental results across vision and language models show that the proposed method can effectively distinguish models derived from a protected source from non-derived models under diverse downstream adaptation settings.

1 INTRODUCTION

Pretrained models are increasingly recognized as valuable digital assets, and the protection of their intellectual property has received growing attention in recent years Pei et al. (2021); Jiang et al. (2026). Regardless of whether a model is open-source or closed-source, its owner may wish to understand how it is reused, adapted, and further derived. Such concerns have already emerged in real-world model development. For example, Anthropic reported large-scale attempts to extract capabilities from Claude through model distillation[1], while ByteDance has also emphasized internally that competing models should not be used as teachers for distillation[2]. These cases highlight a practical model provenance problem: determining whether a downstream model was developed independently or derived from an existing source model.

Early studies in this area focused primarily on verifying model ownership and explored three main directions: The first line relies on trigger-based methods, which encode ownership information into predefined trigger sets or backdoor behaviors for ownership verification Adi et al. (2018); Zhang et al. (2018). A second line explores embedding-based methods, which embed ownership signals into model parameters, intermediate representations, or output behaviors through watermarking or fingerprinting techniques Uchida et al. (2017); Rouhani et al. (2019); Chen et al. (2019); Zhang et al. (2022). A third line investigates structure-based methods, which introduce dedicated model architectures, auxiliary modules, or cryptographic mechanisms to facilitate ownership verification Fan et al. (2022); Mu et al. (2024a). More recently, some studies have begun to investigate model provenance and lineage, with the goal of characterizing derivation relationships among models rather than verifying ownership in isolation. These methods trace how models evolve through fine-tuning, distillation, or other adaptation procedures, and attempt to identify relationships between source models

[1] Anthropic report
[2] Reuters report

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and their descendants Mu et al. (2024b); Wu et al. (2025); Shang et al. (2026); Zhang et al. (2026). A key challenge is that downstream adaptation may substantially change a model's parameters and representations, even though its derivation relationship with the source model remains unchanged. This raises a fundamental question: how can derivation evidence remain verifiable when a descendant model has moved substantially away from its source after adaptation?

To address these challenges, we introduce a new concept, termed the *latent copyright space*, to characterize model derivation relationships in a private low-dimensional space. As illustrated in Fig. 1, downstream fine-tuning may move a model substantially in the original high-dimensional parameter space, while inducing only limited displacement in the copyright space. Consequently, a derived model remains close to its source in this space. This property provides a natural basis for maintaining verifiable derivation evidence throughout downstream adaptation, without requiring prior knowledge of the future fine-tuning.

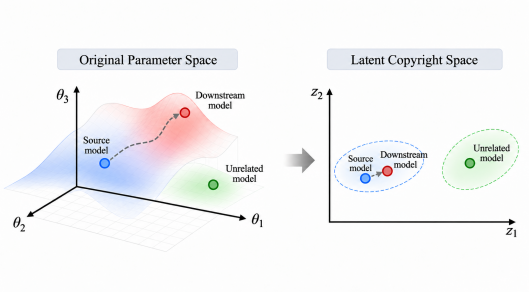


Figure 1: Illustration of the latent copyright space.

Building on this idea, we propose a model provenance framework based on the *latent copyright space*, which we refer to as **LCS**. LCS first uses a secret key to construct a private low-dimensional carrier within the model parameter space. The source model is represented by its coordinates in this space, and a secret copyright identity is assigned to predefined lattice locations. Before release, the model is slightly adjusted so that its copyright representation approaches the target locations while preserving its original task performance. The resulting protected model can then be released and adapted using standard downstream training procedures. During verification, a target model is projected into the same private copyright space and its decoded identity is compared with the registered source identity to determine whether it is derived from the protected model. Extensive experiments across different architectures, downstream tasks, and adaptation procedures demonstrate the effectiveness of LCS for model derivation verification.

Our main contributions are as follows:

- We introduce a new concept of *latent copyright space*, a private low-dimensional space for representing source identity. Instead of directly characterizing model derivation in the original high-dimensional parameter space, the copyright space provides a low-dimensional representation in which derivation relationships can be examined while preserving the model architecture and original task utility.
- Based on this concept, we develop **LCS**, a model provenance framework that encodes source identities as discrete lattice symbols. Before release, LCS adjusts the source model to encode the prescribed identity while maintaining task performance. After downstream adaptation, it verifies derivation by decoding the target model's identity and comparing it with the registered source identity. Notably, LCS requires no prior knowledge of downstream tasks or future adaptation procedures.
- We conduct extensive experiments on vision and language models to evaluate LCS under diverse downstream tasks and adaptation procedures. The results demonstrate that source identity remains verifiable after downstream adaptation. We further discuss a multi-space extension of LCS and potential directions for future research.

2 Related Work

Model Ownership Verification. Model IP (Intellectual Property) protection is a set of technical and legal methods used to secure artificial intelligence and deep learning models from unauthorized copying, theft, fine-tuning, or illegal redistribution. It ensures that creators can prove ownership or restrict access to their valuable AI assets Jiang et al. (2026); Xue et al. (2023); Peng et al. (2023). Typical technical approaches include model watermarking and model fingerprinting. Watermarking methods embed owner-specific information into model parameters, representations Zhang et al. (2022), or behaviors, while fingerprinting methods identify models through their intrinsic func-

tional or behavioral characteristics without explicitly modifying the model Xu et al. (2024). Recent studies have further investigated model lineage tracing, which aims to characterize derivation relationships among models and track their evolution through fine-tuning, distillation, or other model modifications Mu et al. (2024b); Nikolic et al. (2025); Ha et al. (2026). SEAL embeds multi-bit ownership signatures into a latent representation subspace by aligning the hidden representations of selected anchor samples with predefined signature vectors, enabling robust ownership verification under downstream model modifications Dai et al. (2025). Wu et al. (2025) represent LLMs through low-dimensional functional signatures and use their similarity to trace model evolution across fine-tuning, distillation, and other adaptations. Ding et al. (2026) anchor ownership signals into a stable functional subspace to improve robustness against downstream model modifications, while Zhang et al. (2026) extend watermarking to multi-stage derivation chains by preserving contributor-specific signals for lineage and contribution tracing.

Model Watermark. Model watermarking has been widely studied for model ownership verification. Early methods mainly embed ownership information into model parameters or model behaviors Regazzoni et al. (2021); Sun et al. (2023); Chen et al. (2024). Uchida et al. (2017) encode binary watermarks directly into neural network parameters, while Adi et al. (2018) introduce a backdoor-based scheme that verifies ownership through predefined trigger inputs. Rouhani et al. (2019) further embed watermark information into intermediate feature representations, extending watermarking beyond direct parameter manipulation. More recent studies focus on improving robustness under downstream model adaptation. Mo et al. (2025) design task-independent backdoor watermarks that remain verifiable after fine-tuning, and Zhao et al. (2026) exploit contextual sensitivity to construct adaptive watermark triggers for large language models. Overall, existing watermarking methods increasingly emphasize robustness against model modification, but most still formulate ownership verification as the recovery or detection of a predefined watermark signal.

Our proposed method differs from conventional parameter watermarking primarily in how the ownership information is represented and preserved. Existing parameter-based methods typically embed a predefined ownership signal into selected model weights and verify whether the signal remains recoverable after model modification. In contrast, LCS defines a private low-dimensional representation relative to the source model and encodes a key-generated copyright identity using a discrete lattice structure. During downstream adaptation, the copyright representation may drift, while the encoded identity remains recoverable as long as the corresponding decoding regions are preserved.

3 PRELIMINARIES

3.1 PROBLEM FORMULATION

We consider a scenario in which a model owner protects a source model before releasing it for downstream use. Let f_{θ_s} denote the original source model, where $\theta_s \in \mathbb{R}^P$ and P is the number of model parameters. The owner applies a protection mechanism to obtain a protected model $f_{\theta^\star}$, which is then released to downstream users. After release, $f_{\theta^\star}$ may undergo downstream adaptations, such as full-parameter fine-tuning, parameter-efficient fine-tuning, or head replacement, resulting in a target model f_{θ_t}. We say that f_{θ_t} is derived from the protected source model if it is obtained, directly or indirectly, by adapting $f_{\theta^\star}$. Otherwise, f_{θ_t} is considered unrelated. Therefore, given a protected source model $f_{\theta^\star}$ and a target model f_{θ_t}, our goal is to determine whether the target model is derived from the protected source model, i.e.,

$$V(f_{\theta^\star}, f_{\theta_t}) = \begin{cases} 1, & \text{if } f_{\theta_t} \text{ is derived from } f_{\theta^\star}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where V denotes a model derivation verifier.

3.2 LATENT COPYRIGHT SPACE

To characterize model derivation under parameter changes, we introduce a *latent copyright space*, a private low-dimensional representation of model parameters. The motivation is that a large parameter displacement can have a much smaller projection along selected directions. This makes it possible to encode a source identity whose recovery tolerates a certain amount of displacement in

the copyright space. The relevant tolerance depends on the encoding and verification rules: the representation may drift while the encoded identity remains fully or partially recoverable. Accordingly, the mapping provides the coordinates, while identity encoding and verification determine how these coordinates are used to establish evidence of derivation.

Definition 1 (Latent Copyright Space). *Let $\Theta \subseteq \mathbb{R}^P$ be a model parameter space. A* latent copyright space *is a low-dimensional representation space $\mathcal{C} \subseteq \mathbb{R}^d$, where $d \ll P$, equipped with a private mapping $\Phi_K : \Theta \to \mathcal{C}$,*

$$\Phi_K : \Theta \to \mathcal{C}, \tag{2}$$

parameterized by a secret key K. For a model with parameters θ, the vector $\Phi_K(\theta)$ is called its copyright representation.

A desirable copyright space preserves derivation-related information, such that models derived from the same source remain close in their copyright representations, while unrelated models remain distinguishable. We present a concrete construction of Φ_K in Sec. 4.2.

4 METHOD

4.1 OVERVIEW

In this paper, we propose a model provenance framework based on the *latent copyright space*, referred to as **LCS**. The central idea is to represent the identity of a protected source model in a private low-dimensional space constructed from the high-dimensional parameter space. The source identity is encoded as a sequence of discrete lattice symbols in this space. Although downstream adaptation may substantially modify the model parameters, the corresponding displacement in the copyright space can remain limited. This property allows the encoded identity to remain verifiable after downstream adaptation and provides a basis for determining whether a target model is derived from the protected source. **LCS** consists of three stages. First, a secret key defines a private low-dimensional projection matrix and a copyright identity. Second, before release, the source model is slightly adjusted so that its representation in the copyright space is aligned with the prescribed identity while preserving its original utility. The resulting protected model can then be released and adapted through standard downstream training. Finally, a target model is projected into the same copyright space, and its decoded lattice symbols are compared with the registered identity.

4.2 LATENT COPYRIGHT SPACE CONSTRUCTION

We first construct a private low-dimensional space for encoding and verifying the source identity. Let $\theta_s \in \mathbb{R}^P$ denote the parameters of the source model, where P is the number of model parameters. Given a secret key K, we generate a projection matrix [3]

$$C_K = [c_1, \ldots, c_d] \in \mathbb{R}^{P \times d}, \qquad d \ll P, \tag{3}$$

whose columns are orthonormal, $C_K^\top C_K = I_d$. For a model with parameters θ, we define its representation in the latent copyright space as

$$\Phi_K(\theta) = C_K^\top (\theta - \theta_s) \in \mathbb{R}^d. \tag{4}$$

By construction, $\Phi_K(\theta_s) = 0$, so the source model serves as the origin of the latent copyright space. We keep the reference parameters θ_s and the projection matrix C_K fixed throughout encoding and verification.

4.3 COPYRIGHT ENCODING

To encode the copyright identity in the latent copyright space, we use the secret key K to generate a q-ary code

$$s_K = (s_1, \ldots, s_d) \in \mathbb{Z}_q^d, \qquad \mathbb{Z}_q = \{0, \ldots, q-1\}. \tag{5}$$

[3] In our implementation, K initializes a pseudorandom generator to construct random projection directions, which are orthonormalized before encoding and verification.

Each symbol s_i is associated with a set of lattice points along the i-th coordinate:

$$\Lambda_i(s_i) = \{\Delta(\tau_i + s_i + qm) \mid m \in \mathbb{Z}\}, \tag{6}$$

where $\Delta > 0$ is the quantization step and τ_i is a fixed offset determined by K. The integer m indexes different lattice points encoding the same symbol, with adjacent points separated by $q\Delta$.

Since $\Phi_K(\theta_s) = 0$, we select the lattice point nearest to the origin for each coordinate:

$$z_i^\star = \arg\min_{z \in \Lambda_i(s_i)} |z|, \qquad z^\star = (z_1^\star, \ldots, z_d^\star). \tag{7}$$

This choice minimizes the displacement in the copyright space required to encode s_K.

We then adjust the model parameters so that their copyright representation approaches $z^\star$, while maintaining task performance. Keeping θ_s fixed as the reference, we initialize θ at θ_s and minimize

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda \mathcal{L}_{\text{copy}}(\theta), \tag{8}$$

where $\mathcal{L}_{\text{task}}$ is the original task loss, $\lambda > 0$ controls the strength of copyright encoding, and

$$\mathcal{L}_{\text{copy}}(\theta) = \frac{1}{d} \left\| \frac{\Phi_K(\theta) - z^\star}{\Delta} \right\|_2^2. \tag{9}$$

Let $\theta^\star$ denote the resulting protected model. The copyright loss encourages $\theta^\star$ to satisfy $\Phi_K(\theta^\star) \approx z^\star$, while the task loss encourages retention of the source model's task performance. The protected model can then be released for downstream adaptation without changing its architecture.

4.4 Copyright Verification

Given a target model θ_t, we compute its representation using the reference parameters and projection fixed before release:

$$z_t = \Phi_K(\theta_t) = C_K^\top(\theta_t - \theta_s). \tag{10}$$

We decode each coordinate by finding the nearest lattice index and taking it modulo q:

$$\hat{s}_i = \text{round}\Big(\frac{z_{t,i}}{\Delta} - \tau_i\Big) \bmod q, \qquad i = 1, \ldots, d, \tag{11}$$

where $\text{round}(\cdot)$ rounds to the nearest integer, with a fixed rule for ties, and the modulo operation returns a symbol in $\mathbb{Z}_q$. The decoded identity $\hat{s}_K(\theta_t) = (\hat{s}_1, \ldots, \hat{s}_d)$ is then compared with s_K using either exact or soft verification.

Exact verification. Exact verification accepts the derivation claim only when all decoded symbols match the source identity:

$$\mathcal{V}_K^{\text{exact}}(\theta_t) = \mathbf{1}[\hat{s}_K(\theta_t) = s_K], \tag{12}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. Since changing even one decoded symbol causes rejection, exact verification can be sensitive to downstream adaptation.

Soft verification. To accommodate partial identity recovery, we define the symbol agreement score $S_K(\theta_t) = \frac{1}{d}\sum_{i=1}^d \mathbf{1}[\hat{s}_i = s_i]$. Soft verification accepts the derivation claim when this score reaches a threshold:

$$\mathcal{V}_K^{\text{soft}}(\theta_t) = \mathbf{1}[S_K(\theta_t) \geq \eta], \tag{13}$$

where $\eta \in (0, 1)$ is fixed before target evaluation. The threshold can be calibrated on a separate set of unrelated models to meet a prescribed empirical false-positive rate. Both verification rules require only the target parameters and the private reference and encoding information retained before release. Neither requires access to downstream training data or adaptation history.

4.5 Copyright Preservation Analysis

After release, the protected model $\theta^\star$ may undergo downstream adaptation, resulting in a target model $\theta_t = \theta^\star + \delta_{\text{adapt}}$, where δ_{adapt} denotes the parameter update induced by adaptation. Since the reference model θ_s and the projection matrix C_K remain fixed, the corresponding change in the latent copyright space is

$$\Phi_K(\theta_t) - \Phi_K(\theta^\star) = C_K^\top \delta_{\text{adapt}}. \tag{14}$$

Thus, preservation of the encoded identity depends on the magnitude of the adaptation-induced displacement in the latent copyright space.

To characterize the effect of dimensionality reduction, consider a fixed displacement δ independent of C_K. When the columns of C_K span a uniformly sampled d-dimensional subspace of $\mathbb{R}^P$, the standard random-projection result (Dasgupta & Gupta, 2003) gives

$$\mathbb{E}_{C_K}\left[\left\|C_K^\top \delta\right\|_2^2\right] = \frac{d}{P}\|\delta\|_2^2. \tag{15}$$

This follows from $\mathbb{E}_{C_K}[C_K C_K^\top] = (d/P)I_P$. The projected displacement therefore has a root-mean-square magnitude of $\sqrt{d/P}\|\delta\|_2$. This analysis provides a geometric motivation for using a low-dimensional copyright space. Note that the independence assumption is only used for this random-projection analysis and is not required by the proposed verification framework.

For a realized adaptation, identity preservation can be characterized directly through the encoding margin. Under nearest-lattice decoding over all q symbols, adjacent lattice points are separated by Δ, and the decision boundaries lie halfway between them. Assuming that each encoded coordinate is within $\Delta/2$ of its target lattice point, we define the copyright margin as

$$\gamma = \frac{\Delta}{2} - \|\Phi_K(\theta^\star) - z^\star\|_\infty > 0. \tag{16}$$

The complete source identity is preserved whenever $\left\|C_K^\top \delta_{\text{adapt}}\right\|_\infty < \gamma$. Indeed, by the triangle inequality,

$$\|\Phi_K(\theta_t) - z^\star\|_\infty \le \|\Phi_K(\theta^\star) - z^\star\|_\infty + \left\|C_K^\top \delta_{\text{adapt}}\right\|_\infty < \frac{\Delta}{2}. \tag{17}$$

Therefore, each coordinate remains within the decoding interval of its target lattice point, and the encoded identity is preserved. This condition shows that a smaller encoding error provides a larger tolerance to subsequent adaptation in the latent copyright space. A detailed derivation of the projection result and the preservation condition is provided in Appendix A.

5 Discussion

The proposed latent copyright space can be extended to an ensemble verification framework. For a source model θ_s, we construct M private latent copyright spaces using different keys $K_1, \ldots, K_M$. Each space has its own projection matrix, offsets, and identity code, while all identities represent the same source model. Their representations are defined as $\Phi_{K_j}(\theta) = C_{K_j}^\top(\theta - \theta_s), j = 1, \ldots, M$. The source model is jointly optimized to encode all identities:

$$\mathcal{L}_{\text{ens}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \frac{\lambda}{M}\sum_{j=1}^{M}\mathcal{L}_{\text{copy}}^{(j)}(\theta), \tag{18}$$

where $\mathcal{L}_{\text{copy}}^{(j)}$ denotes the copyright loss in the j-th space. This results in a single protected model carrying multiple copyright representations.

For a target model θ_t, each copyright space performs verification independently, and the resulting decisions are aggregated through voting: $\mathcal{V}_{\text{ens}}^{(a)}(\theta_t) = \mathbf{1}\left[\sum_{j=1}^{M}\mathcal{V}_{K_j}^{(a)}(\theta_t) \ge \kappa\right], a \in \{\text{exact}, \text{soft}\}$, where κ is the number of positive votes required for acceptance. This provides a simple extension for improving robustness, which we leave for future work.

6 Experiments

6.1 Experimental Setup

Datasets and models. We evaluate LCS on both vision and language models. For vision experiments, we use CIFAR-100, SVHN, and TinyImageNet datasets, where the source and downstream tasks are constructed from non-overlapping 10-class sets. We use models from the ResNet He et al. (2016) and Vision Transformer (ViT) Dosovitskiy et al. (2021) families as vision backbones.

For language-model experiments, we use models from the Qwen2.5 family and evaluate them on WikiText-2, XSum, and AG News across different language tasks.

Baselines. We compare LCS with four representative model verification methods. **ST-DM** Li et al. (2021) embeds binary information by quantizing projected model weights. **DeepMarks** Chen et al. (2019) embeds user-specific fingerprints into model parameters through a regularization objective. **WM** Uchida et al. (2017) embeds binary watermarks into model parameters using a projection-based regularization loss. **FSW** Ding et al. (2026) embeds ownership signals into a low-dimensional functional subspace of model representations that is selected for robustness to model modifications.

Metrics. We evaluate model derivation verification using symbol agreement, exact recovery rate, true positive rate (TPR), and false positive rate (FPR). Symbol agreement measures the fraction of decoded symbols that match the registered source identity, while exact recovery requires all symbols to be correctly recovered. TPR measures the fraction of derived models correctly accepted by the verifier, whereas FPR measures the fraction of non-derived models incorrectly accepted. We evaluate model utility using classification accuracy for vision tasks and AG News, ROUGE-L for XSum, and perplexity on WikiText-2.

Implementation Details. All experiments are implemented in Python and conducted on NVIDIA RTX 4090 GPUs. For baseline methods, we follow the original papers and publicly available implementations whenever possible, using their recommended hyperparameters. For LCS, we evaluate copyright-space dimensions $d \in [8, 128]$ and quantization steps $\Delta \in [0.025, 0.1]$ and empirically set the soft verification threshold to $\eta = 0.5625$. We provide the basic LCS implementation as supplementary material for review.

6.2 Experimental Results

6.2.1 LCS on Vision Models

Settings. We first evaluate LCS on vision models under different downstream adaptation settings. We use two non-overlapping 10-class subsets of CIFAR-100 as the source and downstream tasks, respectively. To evaluate cross-dataset adaptation, we further adapt the protected models to SVHN and a 10-class subset of TinyImageNet. For ResNet models, we consider full-parameter fine-tuning with SGD or AdamW, LoRA, and fine-tuning after reinitializing the classification head. For ViT models, we evaluate full-parameter fine-tuning with AdamW and LoRA. Unless otherwise specified, we set the copyright-space dimension to $d = 32$, the alphabet size to $q = 4$, and the quantization step to $\Delta = 0.05$. All LoRA experiments use rank $r = 8$. Results are averaged over five downstream adaptation runs with different random seeds.

Results. Table 1 shows that LCS preserves a high level of copyright-symbol agreement under different downstream tasks and adaptation strategies. For ResNet-18 on SVHN, the average agreement remains between 0.9000 and 0.9187, although exact recovery decreases to 0–20%. This indicates that downstream adaptation may alter a small number of copyright symbols without destroying the overall source identity. Under soft verification, the retained symbols remain sufficient for verifying the derivation relationship. On TinyImageNet-10, the copyright identity is recovered under all evaluated adaptation strategies. A similar pattern is observed on ViT, where agreement remains close to 1.0 across all settings. In contrast, independently trained non-derived models exhibit substantially lower agreement, with an average of only 0.1891, and none is accepted by the verifier, corresponding to an empirical FPR of 0%.

Table 1(c) compares LCS with ST-DM and DeepMarks using ResNet-18 under the same CIFAR-100 to SVHN adaptation setting. All models are adapted with full-parameter AdamW and achieve comparable downstream accuracy, ranging from 96.12% to 96.17%. At this comparable level of task utility, LCS retains a symbol agreement of 0.9000 and achieves a TPR of 100%, while ST-DM and DeepMarks obtain agreements of 0.6312 and 0.7484, respectively, with a TPR of 0%. Although the exact recovery rate of LCS is only 20%, the retained copyright symbols are sufficient for soft verification. These results show that LCS better preserves the verification signal under the evaluated cross-dataset adaptation without reducing downstream task utility.

Table 1(d) further evaluates LCS when the protected model undergoes multiple stages of downstream adaptation. The symbol agreement decreases from 1.0000 after the first stage to 0.8750 and 0.8500

Table 1: Verification results of LCS on vision models.

(a) Derived models

Model	Downstream task	Adaptation	Agreement	Exact (%)	TPR (%)	Acc. (%)
ResNet-18	CIFAR-100	LoRA	1.0000	100.0	100.0	76.86
	SVHN	Full SGD	0.9187	0.0	100.0	95.87
		Full AdamW	0.9000	20.0	100.0	96.17
		Reset-head SGD	0.9000	0.0	100.0	95.80
		LoRA	0.9125	0.0	100.0	95.53
	TinyImageNet-10	Full SGD	1.0000	100.0	100.0	68.72
		Full AdamW	1.0000	100.0	100.0	70.56
		Reset-head SGD	1.0000	100.0	100.0	67.20
		LoRA	1.0000	100.0	100.0	64.68
ViT-Tiny	CIFAR-100	Full AdamW	1.0000	100.0	100.0	97.20
		LoRA	1.0000	100.0	100.0	96.12
	SVHN	Full AdamW	1.0000	100.0	100.0	96.90
		LoRA	0.9938	80.0	100.0	96.10

(b) Non-derived models

Model	# Models	Agreement	FPR (%)
ResNet-18	80	0.1891	0.0

(c) Comparison with baselines

Method	Agreement	Exact (%)	TPR (%)	Acc. (%)
ST-DM	0.6312	0.0	0.0	96.12
DeepMarks	0.7484	0.0	0.0	96.16
LCS	0.9000	20.0	100.0	96.17

(d) Successive adaptation

Stage	Downstream task	Agreement	Exact (%)	TPR (%)	Acc. (%)
1	CIFAR-100	1.0000	100.0	100.0	83.74
2	SVHN	0.8750	0.0	100.0	96.15
3	TinyImageNet-10	0.8500	0.0	100.0	68.56

after the second and third stages, respectively. Although exact recovery drops to 0% after the first stage, all derived models continue to pass soft verification, with a TPR of 100% at every stage.

6.2.2 LCS ON LANGUAGE MODELS

Settings. We evaluate LCS on Qwen2.5-0.5B and Qwen2.5-1.5B using XSum for summarization and AG News for text classification. For Qwen2.5-0.5B, we consider LoRA and ProjectionFT on both tasks, together with full-parameter fine-tuning on XSum. For Qwen2.5-1.5B, we evaluate LoRA on both tasks and ProjectionFT on XSum. ProjectionFT updates the attention and MLP projection weights while keeping the remaining parameters frozen. For each setting, the protected model and its corresponding unprotected checkpoint are adapted using the same downstream task, training configuration, and random seed. Models adapted from the unprotected checkpoint are used as parallel negative models. We set $d = 32$, $q = 4$, and $\Delta = 0.05$. LoRA uses rank $r = 8$. For the baseline comparison, we use Qwen2.5-0.5B on XSum with full-parameter fine-tuning for 300 steps at a learning rate of 10^{-4}. Results are averaged over three downstream adaptation runs with different random seeds.

Results. Table 2(a) shows that all derived models retain the complete copyright identity after downstream adaptation, with an agreement of 1.0 and both exact recovery and TPR of 100%. None of the corresponding parallel negative models passes verification, resulting in an empirical FPR of 0%. The same behavior is observed for both model scales and across the evaluated downstream tasks and adaptation methods. The downstream performance of the protected models also remains close to that of models adapted from the corresponding unprotected checkpoints. Table 2(b) reports WikiText-2 perplexity before and after copyright encoding. The measured perplexity changes from 70.5047 to 69.6615 for Qwen2.5-0.5B and from 49.1495 to 49.1174 for Qwen2.5-1.5B. Thus, no increase in perplexity is observed after copyright encoding under this evaluation. Table 2(c) compares LCS with WM and FSW under the same full-parameter downstream fine-tuning setting. At a learning rate of 10^{-4}, LCS retains an agreement of 1.0 and a TPR of 100%. In contrast, WM obtains an agreement of 0.649 with a TPR of 0%, while the two FSW variants achieve agreements of 0.708 and 0.812, with TPRs of 0% and 33.3%, respectively.

Table 2: Verification results of LCS on language models.

(a) Downstream adaptation

Model	Task	Adaptation	Agreement	Exact (%)	TPR (%)	FPR (%)	Task performance	
							Unprotected	LCS
Qwen2.5-0.5B	XSum	LoRA	1.0000	100.0	100.0	0.0	0.2269	0.2257
		ProjectionFT	1.0000	100.0	100.0	0.0	0.2268	0.2303
		Full AdamW	1.0000	100.0	100.0	0.0	0.2394	0.2385
	AG News	LoRA	1.0000	100.0	100.0	0.0	85.00	85.33
		ProjectionFT	1.0000	100.0	100.0	0.0	84.33	85.00
Qwen2.5-1.5B	XSum	LoRA	1.0000	100.0	100.0	0.0	0.2866	0.2803
		ProjectionFT	1.0000	100.0	100.0	0.0	0.2740	0.2683
	AG News	LoRA	1.0000	100.0	100.0	0.0	85.00	85.33

(b) Utility after encoding

Model	Before	After
Qwen2.5-0.5B	70.5047	69.6615
Qwen2.5-1.5B	49.1495	49.1174

(c) Comparison with baselines

Method	Agreement	TPR (%)
FSW (adaptive)	0.708	0.0
FSW (naive top-k)	0.812	33.3
WM	0.649	0.0
LCS	1.000	100.0

6.2.3 Ablation Studies

We study the effects of the copyright-space dimension d and the quantization step Δ using ResNet-18 adapted from CIFAR-100 to SVHN with full-parameter AdamW. We vary $d \in \{8, 16, 32, 64, 128\}$ with $\Delta = 0.05$, and $\Delta \in \{0.025, 0.05, 0.075, 0.10\}$ with $d = 32$. We analyze each factor separately while keeping the others fixed.

Copyright-space dimension. As shown in Fig. 2(a), the projected update ratio increases with d, while symbol agreement remains stable between 0.8719 and 0.9000. All tested dimensions achieve 100% TPR. These results indicate that verification is relatively insensitive to the copyright-space dimension over the evaluated range.

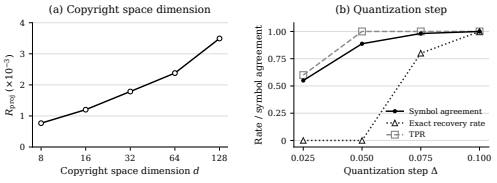


Figure 2: Effects of copyright-space dimension and quantization step.

Quantization step. Fig. 2(b) shows that identity retention is more sensitive to Δ. When Δ increases from 0.025 to 0.10, symbol agreement increases from 0.5500 to 1.0000, while exact recovery increases from 0% to 100%. Meanwhile, downstream accuracy remains between 96.12% and 96.19%. Within the evaluated range, a larger quantization step improves identity retention without noticeably affecting downstream task performance.

7 Conclusion

This paper introduces the concept of *Latent Copyright Space* for model derivation verification under downstream adaptation. Based on this concept, we propose LCS, which encodes a private source identity in a low-dimensional parameter space and verifies whether the identity is preserved in a target model after adaptation. LCS does not require prior knowledge of downstream tasks or adaptation procedures. Experiments on vision and language models demonstrate that the encoded identity can be retained under different downstream tasks, model architectures, and adaptation methods, while non-derived models remain distinguishable. The results on successive adaptation and baseline comparisons further show the applicability of LCS to practical model reuse scenarios. Future work will study larger models, longer adaptation chains, multiple copyright spaces, and a deeper theoretical understanding of identity preservation under downstream adaptation.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, paper organization, and code development. All AI-assisted content and code were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1615–1631. USENIX Association, 2018.

Huajie Chen, Chi Liu, Tianqing Zhu, and Wanlei Zhou. When deep learning meets watermarking: A survey of application, attacks and defenses. *Computer Standards & Interfaces*, 89:103830, 2024.

Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pp. 105–113, 2019.

Yanbo Dai, Zongjie Li, Zhenlan Ji, and Shuai Wang. SEAL: subspace-anchored watermarks for LLM ownership. *CoRR*, abs/2511.11356, 2025.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003. doi: 10.1002/rsa.10073.

Zikang Ding, Junhao Li, Suling Wu, Junchi Yao, Hongbo Liu, and Lijie Hu. Functional subspace watermarking for large language models. *CoRR*, abs/2603.18793, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Lixin Fan, Kam Woh Ng, Chee Seng Chan, and Qiang Yang. DeepIPR: Deep neural network ownership verification with passports. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6122–6139, 2022. doi: 10.1109/TPAMI.2021.3088846.

Anna Yoo Jeong Ha, Wenxin Ding, Stanley Wu, Shawn Shan, Haitao Zheng, and Ben Y. Zhao. Identifying provenance of generative text-to-image models. In *Proceedings of the 35th USENIX Security Symposium*, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Yongqi Jiang, Yansong Gao, Chunyi Zhou, Hongsheng Hu, Si Chen, Anmin Fu, and Willy Susilo. Intellectual property protection for deep learning model and dataset intelligence. *Engineering Applications of Artificial Intelligence*, 163:113024, 2026.

Yue Li, Benedetta Tondi, and Mauro Barni. Spread-transform dither modulation watermarking of deep neural network. *Journal of Information Security and Applications*, 63:103004, 2021.

Weichuan Mo, Kongyang Chen, and Yatie Xiao. TIBW: Task-independent backdoor watermarking with fine-tuning resilience for pre-trained language models. *Mathematics*, 13(2):272, 2025. doi: 10.3390/math13020272.

Xin Mu, Yu Wang, Zhengan Huang, Junzuo Lai, Yehong Zhang, Hui Wang, and Yue Yu. EncryIP: A practical encryption-based framework for model intellectual property protection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21438–21445, 2024a. doi: 10.1609/aaai.v38i19.30140.

Xin Mu, Yu Wang, Yehong Zhang, Jiaqi Zhang, Hui Wang, Yang Xiang, and Yue Yu. Model provenance via model DNA. In *ECAI 2024 - 27th European Conference on Artificial Intelligence*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pp. 1043–1050. IOS Press, 2024b.

Ivica Nikolic, Teodora Baluta, and Prateek Saxena. Model provenance testing for large language models. In *Advances in Neural Information Processing Systems*, 2025.

Jian Pei, Feida Zhu, Zicun Cong, Xuan Luo, Huiwen Liu, and Xin Mu. Data pricing and data asset governance in the ai era. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 4058–4059. ACM, 2021. doi: 10.1145/3447548.3470818.

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

Francesco Regazzoni, Paolo Palmieri, Fethulah Smailbegovic, Rosario Cammarota, and Ilia Polian. Protecting artificial intelligence ips: A survey of watermarking and fingerprinting for machine learning. *CAAI Transactions on Intelligence Technology*, 2021.

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*, pp. 485–497. ACM, 2019. doi: 10.1145/3297858.3304051.

Zhuoyi Shang, Jiasen Li, Pengzhen Chen, Yanwei Liu, Xiaoyan Gu, and Weiping Wang. Attesting model lineage by consisted knowledge evolution with fine-tuning trajectory. In *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, 2026. USENIX Association.

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.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 269–277. ACM, 2017. doi: 10.1145/3078971.3078974.

Zhaomin Wu, Haodong Zhao, Ziyang Wang, Jizhou Guo, Qian Wang, and Bingsheng He. LLM DNA: tracing model evolution via functional representations. *CoRR*, abs/2509.24496, 2025.

Jiashu Xu, Fei Wang, Mingyu Derek 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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 3277–3306. Association for Computational Linguistics, 2024.

Mingfu Xue, Leo Yu Zhang, Yushu Zhang, and Weiqiang Liu. Turn passive to active: A survey on active intellectual property protection of deep learning models. *CoRR*, abs/2310.09822, 2023.

Bingxue Zhang, Xiaofeng Xu, and Feida Zhu. Lineagemark: Multi-user white-box watermarking for contribution tracing in model derivation chains. *CoRR*, abs/2606.17123, 2026.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph. Stoecklin, Heqing Huang, and Ian M. Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security*, pp. 159–172. ACM, 2018. doi: 10.1145/3196494.3196550.

Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4005–4020, 2022.

Gejian Zhao, Hanzhou Wu, Bin Li, Xinpeng Zhang, and Athanasios V. Vasilakos. SensMark: Robust and interpretable model watermarking via contextual sensitivity estimation and adaptive trigger insertion. *Information Sciences*, 756:123869, 2026. doi: 10.1016/j.ins.2026.123869.