Identifying Provenance of Generative Text-to-Image Models

Anna Yoo Jeong Ha, Wenxin Ding†, Stanley Wu†, Shawn Shan, Haitao Zheng, Ben Y. Zhao
† *denotes equal contribution*
Department of Computer Science, University of Chicago
{annaha, wenxind, stanleywu, shawnshan, htzheng, ravenben}@cs.uchicago.edu

Abstract

Fine-tuning provides a fast and cheap way to produce new text-to-image models that are often indistinguishable from ones trained from scratch. Unfortunately, misrepresentation of fine-tuned models creates problems for AI companies and users alike, by disincentivizing competition and misleading users on model quality and ethics of its training process.

In this paper, we propose a model provenance system that identifies models produced by fine-tuning on existing base text-to-image models, using only black-box query access to the models. Our design is informed by analysis showing that one can quantify the feature space difference between text-to-image models by analyzing their responses to detailed prompts. Given a target model, our system analyzes its output, extracts visual features using a generic feature extractor, and compares the distribution against those derived from a pool of base models using Jensen-Shannon divergence. We then apply statistical hypothesis testing to determine if the target model is trained from scratch or fine-tuned, and if the latter, the likely base (parent) model. We evaluate our system across seven popular diffusion models and numerous fine-tuned variants. Our results show high accuracy in attributing model lineage, even under adversarial conditions such as image post-processing or weight perturbations. Finally, we demonstrate real-world efficacy of our system by tracing provenance of in-the-wild models from popular online platforms.

1 Introduction

In the field of text-to-image generative models, the lack of transparency is frustrating users and hindering fair competition. As AI companies continue to unveil new models, little is known about how they are trained and what they are trained on. Exacerbating the issue is the popularization of *model fine-tuning*, where model "variants" can be created from existing base models in a process that is much faster and cheaper than training from scratch [58, 76, 92]. Model fine-tuning is now widely used by both AI companies [7, 77, 78] and academics [47, 48, 75] alike.

Models created by fine-tuning base models are often indistinguishable from base models trained from scratch, and can be advertised as "new" base models. This creates two problems for model trainers and users. First, anyone can fine-tune an existing model and release it as their own "advanced" proprietary model. This is easy given the accessibility of several mature open-source models today. Not only does this confuse users, but it disincentivizes other groups from investing the time and effort to develop and train better models from scratch. Second, given legal and ethical challenges facing models trained on copyrighted content without consent, some new models actively market themselves as "trained from scratch on ethically sourced data" [10, 53, 81, 87]. It is cheap and easy to market a fine-tuned model this way. Indeed, community observations and behavioral similarities suggest this is the case for some of these "ethical" models [9, 30, 55, 86].

To address these challenges, the critical question we must answer is *how to determine if a given generative model was fine-tuned from a known base model, or trained independently from scratch*. Since most deployed models expose only an API or query interface, with no access to weights or training data, a realistic solution must abide by a challenging constraint: query-only (black-box) access to the model in question. Existing solutions fall short: watermarks [20, 31, 91] require modifying the original model during training, impractical for already released models; classifier-based distance comparisons [41] are designed for image classification and fail to capture the high-dimensional, stochastic behaviors unique to generative diffusion models.

In this work, we propose the first black-box provenance identification system for text-to-image diffusion models (see Figure 1). Our approach is driven by an analytical observation: *fine-tuned models retain much of the feature space properties of their parent model, which can be extracted using detailed, generic prompts*. While fine-tuning adapts model weights to new data, it often preserves deeper structural relationships learned during initial training. These residual signals can be probed via generic prompts, extracted and compared. At a high level, our system sends a curated set of detailed, generic

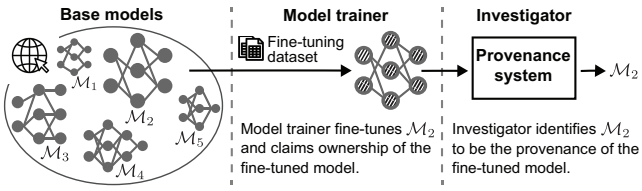


Figure 1: A model trainer fine-tunes a base model and falsely claims ownership. A model provenance system can correctly detect and attribute the model to its original base model.

prompts to the target model, extracts multi-scale features from its image outputs using a visual embedding pipeline, and constructs a high-dimensional distribution of these embeddings. We then compute a mathematical distance (Jensen-Shannon divergence) between this distribution and those generated using the same prompts from a library of known base models. Because fine-tuning has a limited ability to modify the original model's feature space, a fine-tuned model and its parent will exhibit high similarity under our method.

We summarize our key contributions as follows:

- **Black-box provenance algorithm**: Our analysis of diffusion models provides the intuition to formalize the provenance problem as comparing prompt-specific output distributions and guides the selection of test prompts. Leveraging this insight, we develop a black-box model provenance algorithm by measuring and comparing prompt-specific feature distances of models.
- **High accuracy:** We evaluate our system across eight open-source diffusion models and a wide array of fine-tuned variants. Our method correctly attributes fine-tuned models to their base models for all cases in our tests, even after aggressive fine-tuning or architecture modifications.
- **Robust to countermeasures:** We demonstrate strong robustness against 3 countermeasures specifically targeting our system, including model modification via weight-level noise, inference time prompt manipulation, and image post-processing via adversarial perturbations.
- **Real-world efficacy:** We test the efficacy of our system on 6 publicly accessible models with known provenance.

Our work presents a first step towards improving transparency of training methodology in text-to-image models, from the perspective of third parties with only black-box access, such as regulatory agencies. We hope that this will spur additional work that extends black-box model analysis to broader questions and modalities.

2 Background and Related Work

2.1 Text-to-Image Models

We briefly discuss training and fine-tuning for some popular text-to-image models today, including DALLE-3 [4], Imagen [27], FLUX [8], DeepFloyd [2], Stable Diffusion's SD2.1 [80], SD3 [29] and SDXL [65].

Training cost and datasets. State-of-the-art generative models are extremely costly to train from scratch. For example, SD1.5 was reportedly trained on LAION-5B [73], requiring over 150,000 GPU hours and costing $600,000 [79]. Given the intense competition in diffusion model development, companies such as Nvidia [93], StabilityAI [29, 65], OpenAI [4], and others [18, 34, 50, 84] invested heavily in curating proprietary training datasets. These efforts often involve expensive data augmentation techniques to improve data quality, including caption refinement [27] and image filtering [2,29,80]. Ultimately, most base models do not disclose the exact details of their training data.

Fine-tuning and commercial models. Users can fine-tune [39, 70] base models to enhance image quality or optimize performance for specific domains [23, 71, 88]. This approach allows companies to quickly develop customized models by building on existing base models.

Lack of transparency in model provenance. Developers often do not disclose whether a model is original or fine-tuned, and even misrepresent fine-tuned models as being "trained from scratch," either intentionally to avoid ethical or legal scrutiny, or unintentionally due to misunderstanding of the fine-tuning/training process. These claims are hard to verify, given the common practice of commercial models that limit users to black-box access via online APIs [12, 24, 26]. While providers often assert that their models are trained from scratch on ethically sourced data [11, 35, 55, 86], there are currently no reliable tools to verify these claims.

2.2 Related Work

Comparing Deep Neural Networks (DNNs). Prior work compares DNN classifiers by identifying unique "fingerprints" that characterize each classifier's decision boundaries [13, 17, 19, 41, 49, 54, 64, 83]. These fingerprints are used to determine whether a classifier is derived from a parent model [19] or to measure distance between two classifiers [41].

Most existing works focus on curating either pairs of benign inputs and adversarial examples [17, 49, 54, 64] or noise-like adversarial images [19], which are then used as "fingerprinting" inputs to capture and compare model decision boundaries. These generally require white-box access to at least one model in order to generate effective fingerprints. ModelDNA [60] fingerprints a DNN by encoding its training data and input-output information as a compact representation. This per-model representation is then used to train a provenance classifier. However, access to training data is generally infeasible in real-world settings.

Finally, [41] compares DNNs in a black-box setting by approximating each model's decision boundary using linear regression. It trains a regression model to mimic each model's prediction on a set of inputs. The resulting weight vectors are

then concatenated to form a signature for each classifier.

Watermarking DNN classifiers. A related approach involves watermarking DNN classifiers during training, allowing the embedded watermark information to be later extracted for model comparison [83]. However, this method is impractical for our purposes, as existing watermarking techniques are not applicable to generative models. More importantly, it assumes that all generative models are robustly watermarked, a very strong and often unrealistic assumption.

Provenance of generative models. Comparing and tracing generative models is fundamentally more challenging than doing so for DNN classifiers for two key reasons. First, generative models lack explicit decision boundaries, making it harder to characterize their behavior in a comparable way. Second, existing techniques for classifiers rely on deterministic outputs for given test inputs, while generative models produce stochastic outputs.

To identify provenance of generative models, existing works propose crafting adversarial examples using known base models and testing them on target models. This has been applied to both LLMs [37, 42] and diffusion models [38]. However, techniques targeting LLMs have been shown to be unreliable [94], while those designed for diffusion models can be circumvented through grammar refinement or lightweight adversarial training (see §3.2).

Recent work [96] considers the problem of model integrity verification, i.e. using KL divergence of their outputs to verify if a model has been modified (via fine-tuning) from the original. We adapted this approach to our problem setting, and find it insufficient (see §3.2).

Finally, watermarks can be applied to generative models during training [20, 31, 91] or generation time [44]. However, adapting watermarks to address our problem is impractical, because it relies on the strong assumption that all base models are watermarked (similar as discussed above for DNNs).

Data provenance $\neq$ model provenance. Data provenance is a rich research problem with extensive history that is related to our problem in name only. For an excellent summary of data provenance projects, we refer readers to "The End-to-End Provenance Project" [28] and other related literature [3, 63].

3 Threat Model and Potential Solutions

In this section, we introduce the threat model underlying our provenance system, and explore potential solutions and the challenges they face.

3.1 Threat Model

Malicious Model Trainer. We consider a model trainer, either an individual or a company, seeking to obscure the origin of their text-to-image diffusion model $\mathcal{M}_t$. While the trainer fine-tunes a base diffusion model P to produce $\mathcal{M}_t$, they seek to hide the use of P by falsely claiming either $\mathcal{M}_t$ is an independently trained model or fine-tuned from a base model that is not P, or not providing any information.

We assume the model trainer:

- has white-box access to a high-quality base model P;
- has resources to fine-tune P on new data to create $\mathcal{M}_t$;
- can control and customize the fine-tuning process, including modifying training hyperparameters;
- is aware of potential provenance detection systems and may actively attempt to evade detection (discussed in §7).

Provenance Investigator. The investigator seeks to determine whether a model $\mathcal{M}_t$ was trained from scratch or derived from a known base diffusion model via fine-tuning, and if it is fine-tuned, identify its base model.

We assume the investigator:

- has a diverse pool of known, high-quality base diffusion models $\{\mathcal{M}_i\}_{i=1}^{M}$, each could potentially be the origin of $\mathcal{M}_t$; each base model $\mathcal{M}_i$ is verified as trained-from-scratch (discussed in §5); the pool may or may not include P;
- has only **black-box** query access to $\mathcal{M}_t$ and each base model $\mathcal{M}_i$, and can query these models using text prompts;
- has no internal access to these models' architecture, weights, or training data;
- has reasonable computational resources.

Provenance Outcome. The investigator will declare $\mathcal{M}_t$ as "trained from scratch" if $\mathcal{M}_t$ is neither the original nor the fine-tuned versions of any base model from the pool $\{\mathcal{M}_i\}_{i=1}^{M}$. Otherwise, the investigator will label $\mathcal{M}_t$ as "fine-tuned" and output its base model.

3.2 Potential Solutions and Challenges

Our work considers the real-world setting where the investigator has no privileged access to internals or training data of any commercial models. This black-box setting rules out white-box techniques for comparing models, such as comparing model weights [46, 90], watermarking [25, 57], and model DNA [60]. Next, we explore potential solutions (based on existing works) and the challenges they face.

Potential solution 1: fingerprinting generative models using adversarial prompts. Like existing works on classifiers, the investigator can fingerprint each base model in the pool by crafting adversarial examples against it. These are intentionally optimized prompts to mislead the model to produce unintended images. To perform model provenance, this solution must ensure that adversarial examples effective against a base model A will (1) fail to mislead other base models in the pool (i.e., non-transferable to other base models), but (2) successfully transfer to models fine-tuned from A.

This solution faces two key challenges. First, crafting black-box adversarial prompts that does not transfer to other base models is highly challenging, because they must be deeply overfitted into the current model. A recent study [38] proposes a genetic algorithm to iteratively query base models

and discovery dedicated adversarial prompts per base model. We evaluated their findings and found that the supposedly non-transferable adversarial prompt for SD2.1 ("A photo of a cat 3T3t") transfers to several other popular base models (see Figure 10 in Appendix). Second, the target model $\mathcal{M}_t$ can largely reduce the effectiveness of adversarial prompts by applying protection strategies such as rephrasing, grammar refinement, or lightweight adversarial training.

Finally, we experimented with using gibberish text as adversarial prompts to fingerprint each base model, but found that, even without applying grammar check, lightweight adversarial training with random gibberish text already made these prompts ineffective.

Potential solution 2: adapting model integrity check to model provenance. Recent work [96] considers model integrity verification. It determines whether a model has been modified by comparing its output distribution to that of the original model across a set of prompts, aiming to find those that reveal differences. One might consider extending this approach to our problem by verifying $\mathcal{M}_t$ against each base model. However, this can at best verify that $\mathcal{M}_t$ deviates from some base models, but cannot reliably identify the true source.

To verify this, we followed the algorithm of the original paper [96] to verify a fine-tuned SD2.1 model against five popular base models (SD2.1, SDXL, DeepFloyd, Hunyan, and FLUX). It showed that the fine-tuned SD2.1 is identical to DeepFloyd, but deviates from the four other models.

4 Our Proposed Design

The above discussion motivates a deeper examination on the problem of model provenance. At its core, model provenance relies on two factors: 1) the fundamental differences among high-quality, pre-trained base models, and 2) the inherent similarity between a base model and its fine-tuned versions.

Key observations. Our design is motivated by two key observations on these factors. First, training a high-quality generative model from scratch is resource-intensive. This process must construct a high-dimensional visual-textual feature space from the ground up, a space reflects every aspect (and choice) of the training pipeline, including the dataset, model architecture, learning rate, data ordering, optimization dynamics, and more. The learned feature space is a distinct and highly individualized representation, effectively encoding the unique fingerprint of each state-of-the-art base model.

Second, fine-tuning a base model does not reconstruct this space from scratch, but applies small, targeted adjustments to the existing feature structure in response to a limited amount of new data. Much of the original feature geometry remains intact. Therefore, fine-tuned models inevitably retain significant portions of their base model's feature space.

Model provenance by comparing feature spaces. These observations suggest that a practical model provenance system should leverage both the diverse feature spaces of base models and the similarity between fine-tuned models and their bases.

The key challenge, however, is "how to compare these feature spaces across models with only black-box access?" This is difficult for two reasons. First, accurately characterizing a model's feature space is difficult. Existing feature attribution methods, such as saliency maps, SHAP [56], and LIME [68], are designed for classifiers that produce deterministic outputs for each input. In contrast, text-to-image generative models handle long, complex prompts, resulting in a highly intricate prompt-to-image mapping. Second, with only black-box access, investigators cannot control the random seeds used during generation, making the output stochastic even when the exact same prompt is used.

Our proposal: prompt-specific model distance. To address these challenges, we propose to compare models by probing them with *detailed*, *well-supported*, *natural* prompts, allowing us to sample and compare regions of their feature spaces using their output distributions.

Specifically, given a test prompt p, we query the base models and the target model k times, producing k images per model. For each model, the stochastic distribution of its k images represents a sampled region of its feature space defined by p, a well-trained area that captures the model's unique characteristics. These regions are expected to differ meaningfully across different base models while remaining largely unaffected by subsequent fine-tuning. As such, we can perform model provenance on a target model $\mathcal{M}_t$ by computing the prompt-specific distance between $\mathcal{M}_t$ and each of the candidate base models ($\{\mathcal{M}_i\}_{i=1}^{M}$). The base model with the smallest distance to $\mathcal{M}_t$ is the base model of $\mathcal{M}_t$.

Our proposed solution is not arbitrary. It is grounded in a theoretical analysis of model feature spaces and directly addresses the challenges outlined above. This analysis, presented next, provides the intuition to formalize the provenance problem as one of comparing prompt-specific output distributions and guides the selection of effective test prompts.

4.1 Theoretical Insight

Our analysis assumes perfectly trained diffusion models, i.e., those solving the corresponding probability-flow ordinary differential equations [43]. This assumption is idealized. Thus our analytical conclusions are intended to build intuition for designing model provenance in practice and to offer insight on how model distance varies with test prompts, rather than to provide a rigorous analysis of real-world model behavior.

Our analysis measures the distance between two distributions using the Jensen-Shannon divergence [51], due to its advantage of being symmetric and bounded below by 0. We define the distance between two distributions $\mathcal{D}_A$ and $\mathcal{D}_B$ by $J(\mathcal{D}_A, \mathcal{D}_B)$ where $J(.)$ is the Jensen-Shannon divergence.

Comparing diffusion models via output distributions. The behavior of a perfectly trained diffusion model is de-

fined by its input distribution (i.e., training data and process). Thus one can compute the difference between two perfectly trained models as the distance between their input distributions. Next, Theorem 1 further shows that such difference can be measured by comparing their output distributions.

Theorem 1. *Consider two perfectly trained text-to-image diffusion models $\mathcal{M}_A$ and $\mathcal{M}_B$ and a prompt p, let $I(\mathcal{M}_\cdot|p)$ be the input distribution of $\mathcal{M}_\cdot$ conditioned on prompt p and $O(\mathcal{M}_\cdot|p)$ be the output distribution of $\mathcal{M}_\cdot$ when prompted with p, then $J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p)) = J(O(\mathcal{M}_A|p), O(\mathcal{M}_B|p))$.*

This theorem shows that even with black-box access, we can effectively compute the distance between two models by comparing their prompt-specific output distributions, i.e.,

$$Dist(\mathcal{M}_A, \mathcal{M}_B|p) \triangleq J(O(\mathcal{M}_A|p), O(\mathcal{M}_B|p)) \quad (1)$$

The proof for Theorem 1 is in Appendix A.1.

Impact of test prompt p. The following corollary shows that test prompt choice p plays a crucial role in model comparison. For effective comparison, the input distribution conditioned on p should have low variance, as this enables clear and measurable separation between models.

Corollary 1. *Given a prompt p, the prompt-specific distance between two base models $Dist(\mathcal{M}_A, \mathcal{M}_B|p)$ decreases with the variance of $I(\mathcal{M}_A|p)$ and $I(\mathcal{M}_B|p)$.*

This result suggests that natural, detailed prompts are well-suited for measuring prompt-specific distances between models. Natural prompts are non-adversarial with well-defined input distributions, while detailed prompts help reduce variance in these distributions, thereby minimizing randomness (due to random seeds) in the sampled feature regions reflected by the generated outputs. Proof for Corollary 1 is in Appendix A.2.

Identifying provenance of fine-tuned models. Theorem 2 shows that a fine-tuned model has negligible distance to its original base model while having noticeable distance to a different base model. Therefore, we can successfully attribute a fine-tuned model to its base by comparing its distance to candidate base models.

Theorem 2. *Consider two base models $\mathcal{M}_A$ and $\mathcal{M}_B$ and a fine-tuned model $\mathcal{M}_A^F$ from $\mathcal{M}_A$. Using a test prompt p not in the fine-tuning dataset of $\mathcal{M}_A^F$, we have*

$$Dist(\mathcal{M}_{A^F}, \mathcal{M}_A|p) \approx 0 \quad (2)$$

$$Dist(\mathcal{M}_{A^F}, \mathcal{M}_B|p) \approx Dist(\mathcal{M}_A, \mathcal{M}_B|p) \gg 0 \quad (3)$$

This assumes p is not part of the fine-tuning data of $\mathcal{M}_A^F$, a reasonable assumption given the limited size of the fine-tuning dataset. In practice, one can minimize potential error by using multiple provenance test prompts and applying majority vote. Proof for Theorem 2 is in Appendix A.3.

Summary of findings. The above analysis shows that with black-box access, an investigator can effectively compare text-to-image generative models by querying them with detailed, natural prompts and analyzing their prompt-specific output distributions as defined in eq. (1). A fine-tuned model exhibits a low distance from its own base model but a noticeable distance from a different base model. These form the basis of our model provenance system.

It is worth noting that the provenance system can draw test prompts randomly from a very rich pool of detailed, generic, natural text inputs. This flexibility makes provenance tests stealthy and hard to distinguish from normal user queries.

5 Implementing a Provenance System

In this section, we describe a practical implementation of a model provenance system. We begin with a system overview, followed by a discussion of key components facing practical challenges: (1) generating detailed, natural prompts for provenance testing, (2) computing model distance, and (3) handling scenarios where the target model $\mathcal{M}_t$'s base model is not present in the investigator's model pool.

5.1 Overview

The provenance system operates on a pool of m known base models, $\{\mathcal{M}_1, ..., \mathcal{M}_m\}$, with black-box query access to each. The system consists of three sequential components:

1. **Build test prompts** (§5.2) – Given a pool of base models, the investigator constructs a set of prompts for provenance tests. These are detailed, natural prompts that are well-supported by all the base models $\{\mathcal{M}_1, ..., \mathcal{M}_m\}$.
2. **Compute model distance using generated images** (§5.3) – Given a test prompt p, the system queries[1] the target model $\mathcal{M}_t$ and each candidate base model with p for k times, producing k images per model. The system applies a generic, high-performing feature extractor (e.g., S2-CLIP [74]) on each model's k output images to produce their visual embeddings. Next, the system uses these visual embeddings to compute $\mathcal{M}_t$'s distance to each base model. To make this computation tractable, we first apply distance-preserving dimension reduction and then compute the Jensen-Shannon divergence values on the projected embeddings.
3. **Identify the provenance of $\mathcal{M}_t$** (§5.4) – If the base model of $\mathcal{M}_t$ is in the pool, the system will label the model with the smallest distance to $\mathcal{M}_t$ as its base. To handle scenarios where $\mathcal{M}_t$'s base model is not in the pool, our system first applies a z-test [16] to determine whether $\mathcal{M}_t$ is a distinct base model outside the pool (i.e., NULL) or a fine-tuned version of a base model in the pool.

[1]The actual query cost reduces to k if the prompt p has been used in prior provenance tests (i.e., reusing the saved $m \cdot k$ images for the base models).

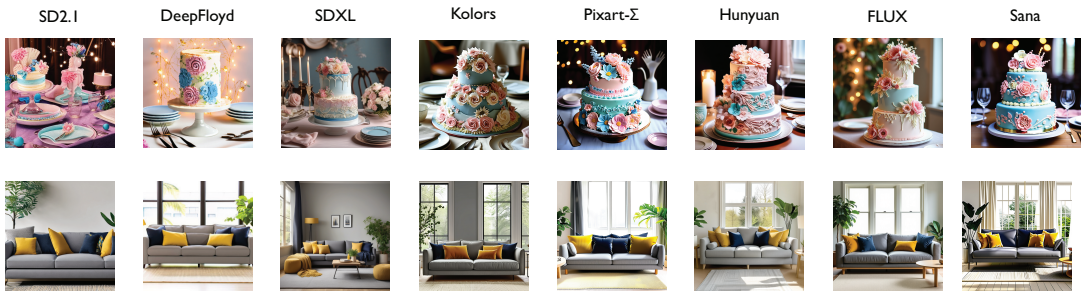


Figure 2: Samples of generated images by querying 8 base models using a detailed prompt about cake (top) and sofa (bottom).

Constructing the base model pool. The provenance system assumes all base models in the pool are high quality, trained from scratch with documented training processes, and excludes models with unknown or undocumented provenance.

We further enforce this requirement by detecting and removing strongly correlated base models from the pool. Specifically, we compute an initial NULL threshold T_{null} using all candidate base models and identify pairs with distance significantly smaller than T_{null}. Such models are either fine-tuned variants of one another or share substantial overlap in training data. For each cluster of highly correlated models, we retain one representative model in the pool. For example, this method confirmed that SD2.0 and SD2.1 are strongly correlated, thus we included SD2.1 in the pool but not SD2.0. Overall, this process produces a pool of distinguishable, uncorrelated base models.

Next, we discuss the individual components in detail.

5.2 Curating Provenance Test Prompts

Our analysis (§4.1) shows that an effective test prompt p should be detailed, natural text prompts that are well-supported by all base models in the pool. This allows the system to sample a well-defined, well-trained feature region per model and use them to separate base models. Also, fine-tuning with limited data is unlikely to change most of these well-trained regions, ensuring similarity between a base model and its fine-tuned versions. This requirement rules out both rare and adversarial prompts that target "under-trained" regions of a model and thus are sensitive to fine-tuning.

Automated test prompt generation. We apply a systematic process to generate test prompts that reference common objects with specific details. First, we randomly select an object from LAION-Aesthetic's frequency list, which contains 23K+ candidate objects ranging from highly popular objects to less common objects. We then query GPT-4 with "generate a photo of `[object]`." As noted in its technical report [5], GPT-4 "upsamples" short captions into detailed descriptions, adding background, composition, setting, and context. The upsampled prompt becomes a test prompt. Note that this is the common method for generating prompts, making the test prompts indistinguishable from common user queries.

Here is a sample prompt on "sofa": "`A modern, stylish gray fabric sofa with soft fabric texture, clean lines, low armrests, three large seat cushions and three back cushions. Neatly arranged mustard yellow and navy throw pillows. Set in a bright, cozy living room with cream rug, light wood coffee table, large sunlit windows in back, and a leafy green plant beside the sofa.`" Additional examples are in Appendix.

For the test prompts on "cake" and "sofa," Figure 2 shows the sample images generated by the eight base models considered by our study. All the base models produce high-quality images matching the detailed descriptions.

Selecting test prompts. For both robustness and stealthiness, the system curates a set of detailed prompts and tests them on all base models in the pool to ensure each will generate intended, high-quality images. To provenance-test a target model, one or more prompts are randomly selected from the pool, and tested on the target model to confirm its suitability. If the model produces inconsistent visual content or low-quality images, an alternative prompt is selected from the pool.

Ineffectiveness of simple/short prompts. We empirically verified that simple, short prompts such as "a photo of sofa" fail to differentiate base models and correlate a base model to its fine-tuned versions (see §6.4). This is because short prompts produce diverse outputs (e.g., with diverse color/background/context), whose distribution is broad and hard to estimate using sampling.

5.3 Computing Model Distance

Given a test prompt p, the system queries each model k times, producing $(m+1)$ sets of k images, where m is the number of candidate base models. To compute the distance between two models, defined by $Dist(\mathcal{M}_A, \mathcal{M}_B|p) = J(O(\mathcal{M}_A|p), O(\mathcal{M}_B|p))$, we first represent each model's k images using a stochastic distribution $O(\mathcal{M}|p)$, then compute the Jensen-Shannon divergence between these distributions.

Converting images to feature distributions. To make the computation tractable, we first apply a high-performing, generic feature extractor to convert each image into a feature embedding. Our current implementation leverages the

recently released S2-CLIP [74] feature extractor. S2-CLIP advances CLIP [67] by incorporating multi-scale feature aggregation. Compared to CLIP and other embedding methods [14, 15], it can better capture both global structures and fine-grained details. Also, S2-CLIP preserves distributional geometry across scales, making it well-suited for our task. For reference, Table 5 in Appendix lists the model distance measured via CLIP and S2-CLIP across models. While both producing the correct provenance conclusion, S2-CLIP further increases the separation between the actual base model and the next best candidate.

Since computing Jensen-Shannon divergence among these embedding sets is still costly, we further reduce them into a lower dimension space using Isomap [85], an advanced, distance-preserving dimension reduction algorithm. Isomap outperforms alternatives such as t-SNE and PCA in our tests. Figure 3 plots a high-level flow of our distance computation module, for a pool of 2 base models and a target model.

Choosing Isomap dimension d. For Isomap dimension reduction, we need to select the target dimension d. Smaller values of d reduce computational cost but can compromise provenance precision due to higher reconstruction loss and reduced model separation. We empirically evaluate the impact of d through reconstruction loss and observe that the loss plateaus when $d > 10$. When $d = 10$, our provenance system can effectively separate the 8 major base models considered in our work. Thus we set $d = 10$ for all of our experiments.

We note that as the pool of base models expands, the provenance system can scale by increasing d or employing a stronger feature encoder. Eventually, the system will reach its capacity limits, where more structured or hierarchical designs will be required. We leave this to future work.

Choosing k. The parameter k defines the number of images generated by a model $\mathcal{M}_A$ per test prompt p, where these k images are used to estimate $O(\mathcal{M}_A|p)$. This is where having a natural, detailed prompt is beneficial, i.e. they constrain the sampled feature region, limiting variance in the feature representations of images generated by a single model.

Based on sampling theory, k directly affects the quality of $O(\mathcal{M}_A|p)$ estimation, with convergence rate approximately $1/\sqrt{k}$, and common practice suggests $k \geq 30$ for distribution estimation. We conducted an ablation study on k and found that while smaller values like $k = 30$ can already correctly identify model provenance, they provide lower confidence, i.e. producing smaller separation margins between the true base model and the next closest candidate. Figure 12 in Appendix shows, for different k values, the distances from a fine-tuned SD2.1 model to its true base model and the next closest candidate. Both distance values stabilize at k increases to 100. Thus we choose $k = 100$ for our experiments.

For scenarios where repeating the same query for k times is a concern for detectability, the investigator can distribute k queries across time, accounts, and IP addresses.

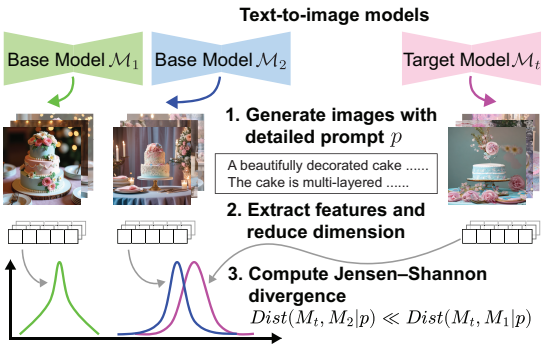


Figure 3: Overview of our model distance computation.

5.4 Provenance via Hypothesis Testing

Using our prompt-specific distance metric, we need to build a mechanism that identifies if a target model $\mathcal{M}_t$ is trained from scratch, or fine-tuned from a base model. For fine-tuned models, we want to identify the base model it was fine-tuned on. Intuitively, if $\mathcal{M}_t$ was trained independently or fine-tuned from an unknown source (not in the pool), its distance to all the candidate bases should ideally be large. Prior work has shown that such group pattern can be detected using z-test [16], a statistical approach to determine how large the distances need to be to suggest this NULL hypothesis. Therefore, our provenance decision includes two steps: first we apply z-test [16] to systematically verify whether the NULL provenance is true or not, and if not, we move to the second step of identifying the parent model from the candidate pool.

Step 1: Detect null provenance via z-test. Given a target model $\mathcal{M}_t$, m base models $\{\mathcal{M}_1, ..., \mathcal{M}_m\}$ and a test prompt p, the system sets M_t's provenance to NULL if

$$\min_{i=1..m} Dist(\mathcal{M}_t, \mathcal{M}_i|p) \geq T_{null}(p) = \mu_p + \tau \cdot \sigma_p \qquad (4)$$

where τ is a parameter, $\mu_p = \frac{2}{m(m-1)} \sum_{i \neq j} Dist(\mathcal{M}_i, \mathcal{M}_j|p)$, and $\sigma_p = \sqrt{\frac{2}{m(m-1)} \sum_{i \neq j} (Dist(\mathcal{M}_i, \mathcal{M}_j) - \mu_p)^2}$. The formulation of eq. (4) follows directly from the z-test process [16].

It is important to note that the NULL threshold $T_{null}(p)$ is *prompt-specific* and *pool-specific*, because both σ_p and μ_p depend on p and the distances among the base models in the pool. On the other hand, once the model pool is given, $T_{null}(p)$ can be pre-computed for each p. Finally, the parameter τ affects the false positive rate of the NULL decision. In theory, under the assumption that m is sufficiently large and $Dist(\mathcal{M}_t, \mathcal{M}_i|p)$ values are i.i.d., $\tau = -1.64$ yields a 5% false positive rate. In this case, $T_{null}(p) = \mu_p - 1.64\sigma_p$.

Step 2: Determine the base model of $\mathcal{M}_t$. If the NULL provenance is not true, i.e., one or more $Dist(\mathcal{M}_t, \mathcal{M}_i|p)$ are below $T_{null}(p)$, we identify the base model as the one with the lowest distance to the target model:

$$\text{Provenance}(\mathcal{M}_t|p) = \arg\min_i Dist(\mathcal{M}_t, \mathcal{M}_i \mid p)$$

6 Experimental Evaluation

We now evaluate our model provenance system using empirical experiments. We configure our experiments to study its feasibility, accuracy and robustness.

- **Effectiveness of the model distance metric**: Using a detailed natural prompt and limited queries, can our model distance metric effectively separate distinct base models?
- **Provenance accuracy**: Can our system accurately locate the base model of a fine-tuned model, or detect a train-from-scratch model? Does it outperform baseline designs?
- **Dependency on prompts**: Does the choice of the test prompt largely affect the provenance performance?

Later in §7 we discuss potential countermeasures where the malicious trainer modifies its fine-tuning to evade detection or cause provenance errors. In§8 we test our system on in-the-wild models, both fine-tuned and trained-from-scratch.

6.1 Experimental Setup

Base models. We consider eight popular base models, covering different architectures (latent vs. pixel vs. DiT), developed by various institutes across multiple countries. With the exception of SD2.1, all models are trained on proprietary datasets with little public information.

- **SD2.1** [80] is a latent diffusion model [69] using a U-Net backbone built by StabilityAI. It is trained on LAION-5B.
- **DeepFloyd IF** [2] is a pixel-based diffusion model from StabilityAI that generates images by first generating smaller images and then upscaling them.
- **SDXL** [65] is a larger and improved U-Net based latent diffusion model from StabilityAI. SDXL is 3x larger than SD2.1 in size and trained on a different dataset.
- **Kolors** [84] is a bilingual latent diffusion model from Kuaishou Technology, a big tech company based in China. It focuses on rendering the text encoder to better understand both English and Chinese prompts.
- **PixArt-Σ** [18] is a diffusion transformer model (DiT) trained by Huawei based in China. It proposes a new attention layer architecture to enable 4k image generation.
- **Hunyuan-DiT** [50] is a DiT model from Tencent based in China. It combines two large text encoders to enable more diverse, accurate and bilingual prompt understanding.
- **FLUX** [8] is a latent diffusion model from Black Forest Labs based in Germany, trained with a DiT architecture. It uses self-attention for denoising instead of convolutions.
- **Sana** [93] is a latent DiT from Nvidia that uses linear diffusion transformers to achieve faster generation.

To our best knowledge, all eight base models are trained independently. They come from six different organizations, often direct competitors, and spread across three countries. Notably, even models from the same company like StabilityAI show major differences: SD2.1 and SDXL employ latent diffusion while DeepFloyd IF is pixel-based; SDXL is 3x larger than SD2.1, and trained on a different dataset, as documented in their technical reports [2, 65, 80]. Note that some base models share training data. For example, SD2.1, DeepFloyd, and Kolors all included LAION-5B.

Fine-tuning setup. We use two datasets for fine-tuning.

- **LAION-Aesthetic** [72]: The LAION dataset consists of image-text pairs scraped from the Internet between 2014 and 2021, using web archives (CommonCrawl). LAION-Aesthetic is a subset of LAION and contains over 1.2 billion image-text pairs with high aesthetic scores according to a CLIP model. It is used to train SD2.1 [80].
- **Photo-Concept-Bucket (PCB)** [66]: This is a high-quality, more recent public dataset from Pexels.com, a curated photo-sharing platform. PCB has minimal overlap with LAION-Aesthetic because Pexels.com blocks CommonCrawl scrapers and its data is collected much more recently. We generate the captions for each image using a generic image captioner (LLAVA [52]).

We follow prior work [45] to fine-tune models. To simulate diverse fine-tuning scenarios, we create fine-tuning sets of varying sizes (100k–500k) and coverage, while also varying training parameters like the number of epochs. Details of the fine-tuning parameters are provided in the Appendix A.5. We focus on fine-tuning two base models, SD2.1 and FLUX, chosen for their distinct architectures and verified, accessible fine-tuning code. Our in-the-wild tests cover more base models (see §8). To ensure fine-tuning preserves generation quality, we evaluate all models using the CLIP Aesthetic score [89]. FLUX achieves an average score of 36.64, with its fine-tuned versions maintaining a comparable score of 36.63. For SD2.1, the average score improves from 27.85 to 31.20 after fine-tuning, indicating enhanced image quality.

Our provenance system. We follow the design in §5.2 to produce detailed, natural prompts on common objects, such as "sofa", "cake", "bassinet". The prompt on "sofa" is listed in §5.2, while others are in Appendix. These test prompts are all well-supported by the eight base models, and lead to a consistent conclusion on provenance performance. By default, we will report the result for using the sofa prompt. We list the results of five different prompts in §6.5.

For each provenance test on a target model, we query the model using the same test prompt p for $k = 100$ times, producing 100 images per model. These queries all follow the standard image generation setting defined by each model.

Computation cost. Our provenance system is computationally light. Using an NVIDIA A100 GPU, generating 100 images on a base model takes between 5 minutes (SD2.1) to 45 minutes (FLUX). The feature extraction process for 100 images takes 3 seconds on the GPU while computing $\mathcal{M}_t$'s distance to each base model takes 2 minutes on a CPU.

Evaluation metrics. For each test prompt, we repeat the provenance test 30 times, and report the mean and standard deviation of the model distance values across the 30 instances. We also study the provenance result across the 30 instances, and find that they are consistently accurate, i.e., 100% accuracy and 0% in standard deviation.

6.2 Effectiveness of Model Distance Metric

We first verify a core assumption of our provenance system: our prompt-specific model distance metric can reliably separate the base models in the pool. Using our default test prompt (on "sofa") as an example, we plot in Figure 4 the pairwise distance between base models in the pool. Note that our distance metric is symmetric, and $Dist(\mathcal{M}_i, \mathcal{M}_i) = 0$. All model pairs show high distance values, ranging from 2.74 to 10.93.

Interestingly, the relative distances among these base models appear to correlate with the high-level methodologies and design choices for developing these models. SD2.1 and SDXL are the closest (2.74), likely due to their architecture similarities. Despite architectural differences (DiT vs. latent), Hunyuan-DiT and SDXL share the same image decoder, which may explain their relatively short distance (3.68). On the other hand, models like PixArt-Σ, FLUX and Sana, exhibit much larger distances from others, due to significant differences in architecture and/or training objectives.

6.3 Provenance Accuracy

To evaluate whether our system can accurately attribute a target model to its base, we examine a diverse set of fine-tuning setups across five scenarios. Later in §8, we also tested six models found online.

Scenario 1. Fine-tuned on original training distribution. We first consider fine-tuning using the same distribution of the base model. Here we fine-tune SD2.1 using a subset of LAION-Aesthetic, the original training distribution of SD2.1. The last column of Table 1 (SD2.1 with 100K LAION) lists the distance between the fine-tuned model and each of the eight base models, in terms of the mean ± standard deviation over 30 runs. The distance between the fine-tuned SD2.1 and SD2.1 is 0.5, nearly 5× smaller than the next closest model (SDXL). Our system consistently identifies it as a fine-tuned version of SD2.1, for each of the 30 runs.

Scenario 2. Fine-tuned on new data distribution. We fine-tune SD2.1 and FLUX on PCB [66], a concept-rich dataset with minimal overlap with LAION [72]. Table 1 shows the model distances for both fine-tuned models (with 100k PCB data). Again, both fine-tuned models are extremely close to their base model (the average distance < 0.5) but well-separated from other bases (the average distance ≥ 3.18).

Scenario 3: Varying fine-tuning parameters. We evaluate the impact of fine-tuning parameters including dataset size, number of training epochs, and learning rate. We find that

Figure 4: Our proposed model distance metric can reliably separate the eight base models. Each cell (i, j)'s value is the distance between model i and j, or $Dist(\mathcal{M}_i, \mathcal{M}_j)$. And $Dist(\mathcal{M}_i, \mathcal{M}_i) = 0$, using a detailed natural prompt on "sofa."

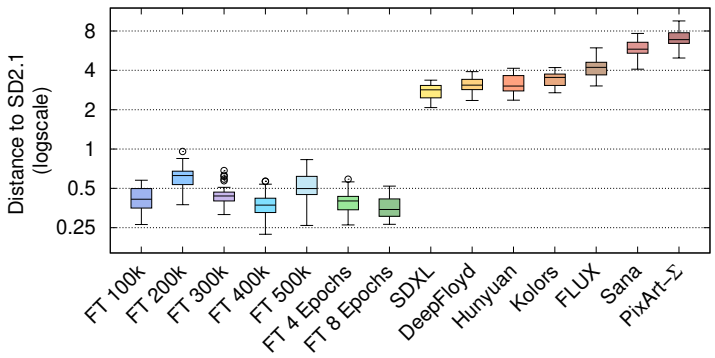
Base Model	Fine-tuned Models		
	SD2.1 with 100K PCB	FLUX with 100K PCB	SD2.1 with 100K LAION
SD2.1	**0.42 ± 0.09**	(4.54 ± 0.68)	**0.50 ± 0.09**
DeepFloyd	3.68 ± 0.64	7.14 ± 1.34	3.66 ± 0.57
SDXL	3.34 ± 0.57	7.31 ± 1.19	(2.40 ± 0.38)
Kolors	4.13 ± 0.44	9.28 ± 1.72	3.47 ± 0.55
PixArt-Σ	6.25 ± 1.09	11.07 ± 2.20	8.51 ± 1.32
Hunyuan	(3.18 ± 0.59)	5.37 ± 1.18	3.22 ± 0.58
FLUX	4.15 ± 0.68	**0.37 ± 0.08**	3.87 ± 0.75
Sana	6.18 ± 0.85	9.87 ± 1.72	5.75 ± 0.84

Table 1: Model distance between fine-tuned models and the candidate base models (using a detailed prompt on sofa). We mark the smallest distance by bold, and the second smallest by ( ). Our system correctly identifies the base model for all three fine-tuned models.

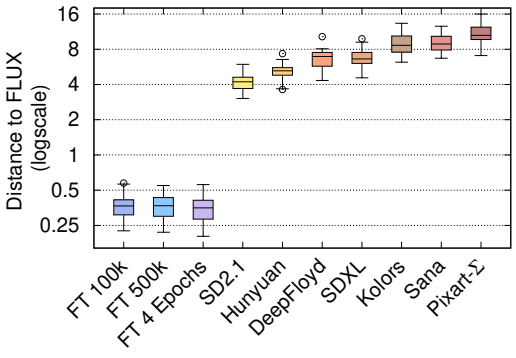
varying the number of epochs (1-8) and varying fine-tuning data size (100k-500k) produce minimum impact on the model distances. Our experiments also cover cases of fine-tuning multiple times, either by using the same data (i.e., increasing the training epochs) or adding 100k new data in each round. As shown in Figure 5a (SD2.1) and Figure 5b (FLUX), the fine-tuned models form a tight cluster around their base model. Finally, when lowering the learning rate from $1e-4$ down to $1e-5$ and $1e-6$, the distance between the fine-tuned model and its base becomes even smaller.

Together, these results highlight the robustness of our provenance system to variations introduced during model fine-tuning. We list the detailed results in Table 4 in Appendix.

Scenario 4: LoRA-based fine-tuning. Compared to full-parameter fine-tuning, lightweight adaptations like LoRA make smaller modifications to a model's feature space, making them easier to identify using our system. We trained SD2.1 LoRAs using the default configuration from HuggingFace,



(a) Distances between SD2.1 and its fine-tuned versions << those between SD2.1 and other 7 base models.



(b) Distances between FLUX and its fine-tuned versions << those between FLUX and other 7 base models.

Figure 5: Our provenance system is robust against variations in hyperparameter used for fine-tuning. These models are fine-tuned on PCB, with varying size (100k-500k) and training epochs (4 or 8). The resulting fine-tune models form a tight cluster around their base models, i.e., their distance to the base model, SD2.1 in (a) and FLUX in (b), remains close to 0.

producing four style mimicry models trained on 20-30 artwork per artist. As expected, our provenance system produced accurate results on LoRAs. Later in §8, we tested two popular LoRAs found online (a SDXL LoRA and a FLUX LoRA). The result remains consistent.

Scenario 5: Trained-from-scratch models. We consider the scenario where the target model is a distinct base model (i.e., train-from-scratch) not covered by the pool, or a fine-tuned version of it. For these models, our provenance system should output "NULL." For our test, we remove SD2.1 from the model pool and make it the target model $\mathcal{M}_t$. Using the new pool of seven models (without SD2.1), the NULL threshold is $T_{null} = 1.59$. The minimum distance of SD2.1 to the seven base models $\min_{i=1..7} Dist(\mathcal{M}_t, \mathcal{M}_i|p) = 3.12 > T_{null}$, producing a "NULL" result. We repeat the test using fine-tuned versions of SD2.1 and obtain the same "NULL" result.

6.4 Comparison to Baseline Designs

Our system made several key design choices: using detailed, natural prompts, using S2-CLIP to extract features. We now evaluate whether these choices outperform baseline methods.

FID-based measures. One might consider using FID scores to compare model outputs under the same prompt. However, while FID is widely used to assess generation quality, it proves ineffective for provenance detection. It fails to distinguish between base models or link them to their fine-tuned versions. This is because FID measures distributional distance in Inception feature space, which captures broad visual statistics rather than the subtle, model-specific generation patterns that persist through fine-tuning [40].

S2-CLIP vs. CLIP. While both capture semantic similarities, S2-CLIP is better at capturing fine-grained distributional differences crucial for provenance detection. Table 5 in Appendix lists the model distance measured via CLIP and S2-CLIP. While both producing the correct provenance conclusion, S2-CLIP increases the separation between the actual base model and the next best candidate, i.e., 8× separation margin compared to CLIP's 5× margin.

Using simple text as test prompts. We empirically verified that simple prompts (e.g., "`photo of a sofa`") are unreliable for provenance tests, under FID, CLIP or S2-CLIP based metrics. They produce images of high variation in color, background and style, making it hard to capture subtle, model-specific details using limited queries (see Figure 11 in Appendix). In contrast, detailed prompts constrain outputs and emphasize distinctive features learned by each model, providing key signals for provenance detection.

6.5 Impact of Test Prompt Choices

Our analysis (§4.1) suggest that detailed natural prompts should offer strong provenance results. Thus a practical provenance system can draw test prompts from a rich pool. Since our design uses prompts about common objects, we examine whether object choice impacts provenance results.

Intuitively, an object that appeared more frequently in the model's training data would lead to a stronger provenance result, because its corresponding feature region is well-defined. Since SD2.1 is trained on LAION-Aesthetic, we select five objects from LAION-Aesthetic with different ranks in frequency: "sofa" (rank #109), "cake" (#158), "camp" (#663), "blazer" (#1115), and "bassinet" (#10800). We use each to create a detailed, natural prompt for provenance testing.

We set the target model $\mathcal{M}_t$ as a SD2.1 model fine-tuned using 100k PCB. Table 2 lists model distances between $\mathcal{M}_t$ and the base models, for each of the five prompts. All five prompts lead to the same provenance outcome, correctly identifying SD2.1 as the base model. The fine-tuned model, when

Base Model	Fine-tuned SD2.1 with 100K PCB				
	Prompt on **"sofa"** Ranked 109th	Prompt on **"cake"** Ranked 158th	Prompt on **"camp"** Ranked 663th	Prompt on **"blazer"** Ranked 1,115th	Prompt on **"bassinet"** Ranked 10,800th
SD2.1	**0.42 ± 0.09**	**0.55± 0.13**	**0.50 ± 0.12**	**0.51 ± 0.19**	**0.42 ± 0.09**
DeepFloyd	3.68 ± 0.64	(3.67 ± 0.57)	(2.39 ± 0.54)	3.53 ± 0.57	1.87 ± 0.39
SDXL	3.34 ± 0.57	4.93 ± 0.89	3.16 ± 0.52	(2.59 ± 0.39)	(1.33 ± 0.23)
Kolors	4.13 ± 0.44	7.94 ± 2.33	17.82 ± 2.92	8.91 ± 1.47	6.70 ± 1.65
PixArt-Σ	6.25 ± 1.09	13.76 ± 3.18	5.80 ± 1.12	10.46 ± 2.52	6.95 ± 1.64
Hunyuan	(3.18 ± 0.59)	8.57 ± 2.23	6.06 ± 1.07	10.69 ± 2.46	2.77 ± 0.29
FLUX	4.15 ± 0.68	7.88 ± 1.50	9.23 ± 1.91	8.64 ± 1.56	4.17 ± 0.65
Sana	6.18 ± 0.85	13.90 ± 2.63	9.53 ± 1.63	14.03 ± 2.33	7.85 ± 0.98

Table 2: Distance from the target model (a fine-tuned SD2.1 using PCB) to the eight base models, using test prompts crafted on different objects. For each object, we also show its rank in terms of frequency in the LAION-Aesthetic dataset used to train the SD2.1 base model. All five prompts produce the correct provenance result.

probed with different prompts, remains very close to its base model (distance <0.5). The key difference among the five prompts is that the distance gap to the next closest model slowly decreases as the object becomes less "popular." This is particularly visible for "bassinet" (#10800), where the next closest model (SDXL) is 1.33 from the target model, compared to that of the base (0.42). This is expected – objects less represented in the base model are more susceptible to changes in their feature representation during fine-tuning.

Test prompt *p* overlapping with fine-tuning data. Another interesting finding is on the prompt related to "cake", because "cake" is a popular object in the fine-tuning dataset (100k PCB), with 288 samples. Yet our system reliably attributes the fine-tuned model to its base, likely because limited fine-tuning data fails to override original training distributions and alter SD2.1's feature space.

7 Countermeasures

In this section, we explore potential countermeasures that model practitioners may employ to avoid successful model provenance detection. Since the provenance system characterizes each model by querying it using test prompts and analyzing the output distribution, knowledgeable adversaries can potentially cause misrepresentation of models by exploiting the provenance methodology, i.e. comparing output distributions, and/or specific knowledge of test prompts.

For methodology-based countermeasures, we consider two potential directions:

- Modifying model internals, with enough impact such that the model will be meaningfully different from its parent. Thus its image distribution produced by our provenance tests will associate it to a different base (see §7.1);
- Disrupting the provenance test process, such that the image distribution produced by the model will appear sufficiently different from the parent (see §7.2).

For knowledge-based countermeasures, targeting specific test prompts is infeasible because our provenance test does not rely on fixed prompts but draws prompts randomly from a large pool of generic prompts, derived from more than 23,000 objects in LAION-Aesthetic. While theoretically an attacker could fine-tune the model against the entire generic prompt pool, doing so would require training costs comparable to, or exceeding, training a model from scratch, while preserving model utility. Thus run-time detection of provenance queries is the only practical alternative, which we evaluate in §7.3.

Our findings. Among the evaluated countermeasures, only post-inference image perturbation may produce a noticeable impact on the measured model distances. However, even at high perturbation budgets, which resulting in significant image distortions, this method is still insufficient to mislead the provenance system.

7.1 Modifying Model Internals

The most direct way to make a fine-tuned model look different from its parent model, is to actually modify its feature distributions so it becomes meaningfully different from that of the parent. Recall that a key assumption of our work is that fine-tuning is generally incapable of significantly altering the feature space defined by the initial pre-training process. The question remains, how can a model trainer meaningfully change the model's internal features without a significant and costly training process.

Altering weights then fine-tune. Model trainers can directly alter model weights in an attempt to produce change in the generated image distribution. Our tests show that simply adding random gaussian noise has little impact on provenance detection, but quickly degrades model performance. Instead, we hypothesize that adding significant noise followed by additional fine-tuning might produce a functional model that evades detection.

Here, we add a wide range of gaussian noise ($\sigma = 0.004 \rightarrow 0.016$) to the weights of a pretrained SD2.1 model and then applying additional fine-tuning using 100k images. Results in Figure 6 show that this countermeasure succeeds only after gaussian noise $\sigma \geq 0.016$. While this fine-tuned countermea-

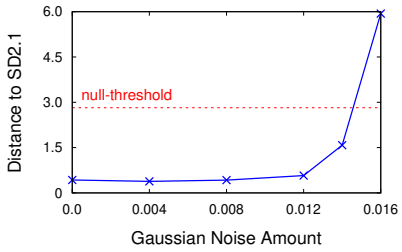


Figure 6: Adding gaussian noise to SD2.1 model weights before fine-tuning is ineffective below 0.016 at preventing successful detection of model provenance (distance to SD2.1 is below null-threshold).

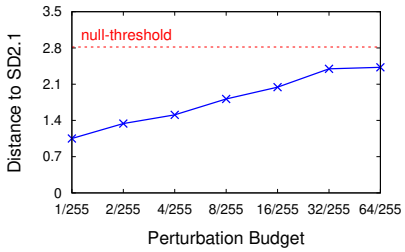


Figure 7: Adversarially perturbing output images to disrupt the S2-CLIP feature extractor cannot successfully prevents detection of model provenance even using high budgets $\varepsilon = \frac{64}{255}$.

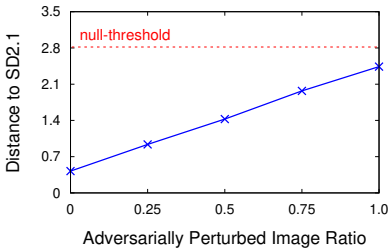


Figure 8: Mixing perturbed images with normal output images further degrades the effectiveness of the countermeasure. We use perturbed images with the highest perturbation budget $\varepsilon = \frac{64}{255}$.

Figure 9: Examples of degraded image quality when a fine-tuned version of SD2.1 is perturbed sufficiently to evade provenance detection (prompt for "sofa," $\sigma = 0.016$).

sure is better than adding raw noise alone, the resulting image degradation is still very significant (see examples in Figure 9), making the countermeasure infeasible in practice.

Other model weight modification methods. We note that we have not explored more "targeted" methods to modify model weights apart from randomized weight modifications. For example, a trainer could try to make their model's weights more similar to another known base model. However, we are not aware of any ways to achieve this without retraining the model or significantly degrading model performance.

7.2 Tampering with Provenance Queries

If we do not modify the model itself, the alternative is to disrupt the test queries from the provenance system so that they produce an erroneous result. To disrupt these queries, a model trainer has two options. They can either alter the prompts from the provenance system before they are processed by the model, or they can let the model process the correct query but alter the resulting images. In the following experiments, we consider both potential countermeasures:

- modifying the input text prompt before image generation
- performing adversarial perturbation attacks on generated images to directly confuse the feature extractor used by the provenance system.

Modifying input texts is ineffective. We first try intercepting and modifying queries before they are answered by the model. To preserve the semantic meaning of the query, we rewrite queries by paraphrasing them with Llama3.1 [36]. Even when each query is paraphrased with the highest amount of randomness (temperature = 1.0), the distance between a fine-tuned SD2.1 model and pretrained SD2.1 using the detailed sofa prompt only increases from 0.42 to 0.59, still well below our null threshold of 2.82 for sofa. We conclude that prompt paraphrasing is unable to impact provenance results.

Adversarially perturbing image outputs is ineffective. Now we consider a scenario where the model trainer modifies images before they are sent back to the provenance system. We first considered a weak attack that adds Gaussian noise to the image. Quick experiments confirm that it has no impact on provenance detection. We then consider a stronger countermeasure, which modifies images by adding noise specifically computed to disrupt the downstream feature extractor in model provenance. Here, we assume a strong attacker (model trainer) who knows that S2-CLIP will be used to calculate features for provenance testing, and use it to compute noise optimized to confuse the feature extractor. We discuss this and other real-world constraints later in §7.3.

We design an *untargeted* adversarial attack with the goal of scattering the distribution of image features in as many random directions as possible. We implement this attack using standard PGD with the goal of maximizing a shift in the generated image's features in S2-CLIP. We solve Eq. 5 for each generated image before returning to the user, with the generated image x, feature extractor F, optimized noise δ, and perturbation budget ε.

$$\max_{\delta} F(x+\delta) - F(x) \text{ subject to } \|\delta\|_\infty \leq \varepsilon \tag{5}$$

We test this adversarial perturbation countermeasure using a range of different perturbation budgets, and plot results for the sofa prompt in Figure 7. For a fine-tuned SD2.1 model, adding increasingly strong feature perturbations to images

still fail to push JS divergence values above the null-threshold. Even at a very large $\epsilon = \frac{64}{255}$ perturbation budget, the model can still be identified as a fine-tuned model. Other prompts show similar results.

Despite its failure, this is by far the most promising countermeasure approach we have found in our tests. Next, we discuss some real world constraints that will significantly impact this and similar countermeasures in practice.

7.3 Real World Considerations

Finally, we consider potential constraints in real world deployment of countermeasures. We envision a scenario where an investigator has online query (black-box) access to models they are evaluating, and can issue queries over a period of time. The challenge for the model trainer, is that they must deploy any countermeasures carefully to minimize negative impact on customer queries while ensuring they affect most or all of the investigator's queries.

Detecting provenance queries is difficult. Since model provenance relies on generating image using the same prompt multiple times, companies can try to detect repeat queries as potential provenance queries. However, repeated queries using the same prompt is a key component to the workflow of many text-to-image users [21], with companies like Midjourney even having developed explicit features to generate multiple images for the same prompt [59]. In addition, an investigator can make this more challenging by spreading provenance queries for a model over time, over user accounts, and over multiple IP addresses. Finally, recall that provenance queries prefer popular concepts that have many training samples in the parent model's training dataset (§6.5). This means provenance queries are likely to match popular prompts with high volumes of benign user queries, making identifying them similar to finding needles in a haystack.

A model trainer's best chance of evading provenance detection is applying adversarial perturbations to images generated by what they suspect might be provenance queries. The negative visual impact of perturbations means they will not be used on all queries. In Figure 8, we show the impact on potency of the countermeasure if only a portion of the provenance queries are identified and altered. A model trainer would have to catch all provenance queries to even come close to possibly altering the result.

8 Model Provenance in the Wild

Finally, we assess the efficacy of our system on models in the wild. We tested models with public (black-box) query access and clear training documentation stating whether they were trained from scratch or fine-tuned from a base model.

In-the-wild models. We looked for models in two popular model hosting sites: HuggingFace [30] and CivitAI [23]. Our goal was to find popular models with clear and definitive documentation on their training methodology. We identified the following six models for our tests.

- **FFUSION [32]:** One of the most popular CivitAI SD2.1 fine-tunes. It is fine-tuned on an unknown dataset for 24,000 steps (120 epochs). Assuming a high batch size of 4096, we can deduce the fine-tuned dataset is $\approx$ 800k.
- **Waifu-Diffusers [61]:** A popular HuggingFace model fine-tuned on SD2.1 using 110k anime-style images.
- **bigASPv2 [62]:** An SDXL derivative on CivitAI that was fine-tuned on a dataset of 6.7 million photo realistic images, over 19,531 steps (6 epochs), with batch size 2048 and learning rate 1e-4.
- **CiroN2022toyface [22]:** A SDXL LoRA model that generates "toy face" style images.
- **FluxSuperReaslismLoRA [82]:** A FLUX LoRA model fine-tuned with 55 images to generate extremely realistic, photograph-like images.
- **CogView3 [97]:** A model trained from scratch using LAION-2B and a small internal dataset, for a total of $\approx$ 900,000 steps with a batch size up to 2048.

Similar to our previous experiments, we generate 100 random images per model 30 times (trials), to compare against our 8 base models. We use the detailed "cake" prompt.

In-the-wild provenance results. Results in Table 3 show that our provenance method correctly identifies both FFUSION and Waifu-Diffusers as SD2.1 fine-tunes, CiroN2022toyface as SDXL fine-tunes, and FluxSuperReaslismLoRA as FLUX fine-tunes. In fact, their base model is the only one below the null-threshold of 3.16, suggesting a highly confident result. The results are slightly more complicated for bigASPv2, where four base models show distance below the null-threshold. The results clearly identify bigASPv2 as a fine-tuned model, and SDXL (the real parent model) as the second best candidate.

For these five fine-tuned models, we observe that the key difference between bigASPv2 and the others is the magnitude of the fine-tuning that produced the models. Recall that the premise behind our work is that fine-tuning operations are limited in their ability to alter the feature space defined by a model's pretraining stage. The two LoRA models used very limited data; FFUSION and Waifu-Diffusers were fine-tuned on datasets with less than 1 million image/text pairs. This relatively small fine-tuning dataset left much of the original SD2.1 feature space intact, making it easy to identify SD2.1 as the only possible parent model. In contrast, bigASPv2 was fine-tuned on a dataset more than 6x larger (6.7 million). This may be the reason why our system is able to identify bigASPv2 as a fine-tuned model, but unable to clearly identify the single parent model. We believe that large scale fine-tuning can reshape enough of the feature space to make it resemble multiple base models, but not enough to make it as distinctive as a model trained from scratch. Finally, we note that there

Base Models	In the Wild Models					
	FFUSION (SD2.1)	Waifu-Diffusers (SD2.1)	bigASPv2 (SDXL)	CiroN2022toyface (SDXL LoRA)	FluxSuperReaslismLoRA (FLUX LoRA)	CogView3 (from scratch)
SD2.1	**0.63 ± 0.14**	**1.99 ± 0.63**	4.80 ± 0.73	3.33 ± 0.56	5.11 ± 0.71	8.67 ± 1.49
DeepFloyd	5.78 ± 1.22	4.88 ± 1.53	3.10 ± 0.43	4.25 ± 0.96	6.73 ± 1.22	3.59 ± 0.72
SDXL	3.56 ± 0.54	6.58 ± 1.65	**2.82 ± 0.49**	**0.69 ± 0.15**	7.43 ± 1.26	6.50 ± 1.26
Kolors	8.41 ± 1.69	9.58 ± 3.87	3.52 ± 0.34	5.14 ± 0.95	9.41 ± 2.54	9.98 ± 2.79
PixArt-Σ	13.27 ± 2.96	9.40 ± 3.64	**2.65 ± 0.57**	8.94± 1.95	11.95 ± 2.07	4.92 ± 1.20
Hunyuan	7.72 ± 1.85	14.90 ± 4.88	3.53 ± 0.46	6.11 ± 1.29	5.64 ± 1.10	15.77 ± 3.07
FLUX	8.21 ± 1.35	11.73 ± 3.35	2.98 ± 0.50	9.18 ± 1.85	**0.46 ± 0.09**	10.18 ± 1.81
Sana	13.35 ± 3.04	15.12 ± 4.94	5.70 ± 0.71	7.85 ± 1.38	9.68 ± 2.64	7.24 ± 1.37

Table 3: Distance to base models, for five fine-tuned models and one train-from-scratch model using the cake prompt. Our system correctly identifies FFUSION and Waifu-Diffusers as fine-tuned on SD2.1, CiroN2022toyface as SDXL fine-tunes, FluxSuperReaslismLoRA as FLUX fine-tunes, and bigASPv2 as fine-tuned with candidates SDXL (real parent) and PixArt-Σ. CogView3 is correctly flagged as trained from scratch, with all model distances above the NULL threshold.

are architectural relationships that may explain similar scores between SDXL, PixArt-Σ, FLUX, and DeepFloyd[2].

Finally, CogView3 is a model that is documented to be trained from scratch [97]. Its distance to all of our base models exceed the null-threshold, confirming that our method can also identify if models are actually trained from scratch.

9 Conclusion

Today, text-to-image models lack transparency, with no reliable way to verify training claims. As the AI ecosystem evolves, key stakeholders are in dire need of mechanisms to test and validate claims of model training methodology. Our work provides a first step towards addressing this need, by introducing a black-box approach for estimating feature space similarity between diffusion models using only their image output. We demonstrate that our system performs well across diverse training setups, against countermeasures, and through in-the-wild experiments on publicly deployed models.

In ongoing work, we plan to study and quantify the different levels of impact on our distance metrics from advanced (and stronger) fine-tuning techniques, and use those results to refine and improve our distance estimation techniques. With improved sensitivity, we hypothesize that it might be possible to reliably distinguish the impact of specific commonalities between models, i.e. shared architectural components or subsets of overlapping training data. We also hope to extend our approach to other generative model architectures.

Acknowledgement

We thank our anonymous reviewers for their insightful feedback. This work was supported in part by the NSF grant CNS-2241303 and ONR grant N000142412669. Stanley Wu was supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2140001. Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

[2]PixArt-Σ was fine-tuned using SDXL's VAE, FLUX was trained by former stability.ai team members who released SDXL, and DeepFloyd was also developed by stability.ai

Ethical Considerations

In this work, we developed a practical tool to enable third parties to determine a model's provenance using only black-box query access. In regulatory and legal settings, our tool is designed to identify potential false claims and assist early-stage investigations. While results can inform and justify subsequent legal requests, such as access to model weights, training data or training logs, they are not meant to serve as definitive proof on their own. We hope this makes forward progress in addressing a gap in transparency that has limited efforts to regulate today's generative AI systems.

Our research may expose undisclosed training practices potentially leading to reputational harm if models are found to be fine-tuned from controversial base models without proper acknowledgement. However, our work also provides valuable tools for companies to verify their own model provenance and ensure transparency in their training practices. Furthermore, better transparency about model origins helps users make more informed decisions when using AI tools, especially in light of rising concerns regarding ethical training pipelines and potential biases inherited from base models.

While our tool is designed for investigative purposes, we acknowledge potential adversarial use. One concern is whether the provenance result of a model could facilitate adversarial attacks against it. For example, when an attacker determines that a target model $\mathcal{M}_t$ was fine-tuned from a known base model, they may directly apply or adapt adversarial examples or vulnerabilities known to the base model to attack $\mathcal{M}_t$. On the other hand, the attack transferability from the base model to its fine-tuned versions is uncertain. Understanding how model provenance outcomes affect adversarial transferability is an important direction for future work.

Finally, model provenance tracking requires querying models to collect samples of model generation behavior. We en-

sure that our evaluation does not violate any terms of service or usage policies.

Open Science

We have made all the artifacts necessary for evaluating our contributions publicly available. We release complete implementation of our provenance system including image generation, model distance computation and statistical testing. Code is available at: https://zenodo.org/records/17870201.

References

[1] Aditya Belekar. Guide to training and fine-tuning Flux.1. https://blog.segmind.com/guide-to-training-and-fine-tuning-flux-1/, 2024.

[2] Stability AI. Stability AI releases DeepFloyd IF, a powerful text-to-image model that can smartly integrate text into images. https://stability.ai/news/deepfloyd-if-text-to-image-model, 2023.

[3] Abdullah Mujawib Alashjee, Salahaldeen Duraibi, and Jia Song. Dynamic taint analysis tools: A review. *Proc. of IJCSS*, 13(6), 2019.

[4] James Betker et al. Improving image generation with better captions. https://cdn.openai.com/papers/dall-e-3.pdf, 2023.

[5] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions, 2023. OpenAI Technical Report.

[6] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. 2006.

[7] Black Forest Labs. Flux 1.1 pro: Better and faster image generation. https://flux1.ai/flux1-1, October 2024.

[8] Black Forest Labs. FLUX.1-dev model card. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024.

[9] Blunge.ai. https://www.blunge.ai.

[10] Rishi Bommasani et al. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*, 2023.

[11] BRIA AI. BRIA v2-3 model card. https://huggingface.co/briaai/BRIA-2.3/, 2024.

[12] Canva. Magic Studio AI image generator. https://magicstudio.com/ai-image-generator/, 2023.

[13] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proc. of AsiaCCS*, 2021.

[14] Mathilde Caron et al. Emerging properties in self-supervised vision transformers. In *Proc. of ICCV*, 2021.

[15] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Proc. of NeurIPS*, 33, 2020.

[16] George Casella and Roger Berger. *Statistical inference*. CRC press, 2024.

[17] Jialuo Chen et al. Copy, right? a testing framework for copyright protection of deep learning models. In *Proc. of IEEE S&P*, 2022.

[18] Junsong Chen et al. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024.

[19] Yufei Chen, Chao Shen, Cong Wang, and Yang Zhang. Teacher model fingerprinting attacks against transfer learning. In *Proc. of USENIX Security*, pages 3593–3610, 2022.

[20] Yunzhuo Chen, Jordan Vice, Naveed Akhtar, Nur Al Hasan Haldar, and Ajmal Mian. Image watermarking of generative diffusion models. *arXiv preprint arXiv:2502.10465*, 2025.

[21] Kevlin Chow. Getting starting with ai image generation and prompt engineering. https://adeptdept.com/blog/getting-starting-with-ai-image-generation-and-prompt-engineering/, 2024.

[22] CiroN2022. Ciron2022 toy face. https://huggingface.co/CiroN2022/toy-face, 2022.

[23] Civitai. What the heck is Civitai? https://civitai.com/content/guides/what-is-civitai, 2022.

[24] Craiyon. Craiyon AI image generator. https://https://www.craiyon.com/, 2022.

[25] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proc. of ASPLOS*, 2019.

[26] DeepAI. DeepAI AI Image Generator. https://deepai.org/machine-learning-model/text2img, 2017.

[27] Google Deepmind. Imagen 3. https://deepmind.google/technologies/imagen-3/, 2024.

[28] Aaron M. Ellison, Emery R. Boose, Barbara S. Lerner, Elizabeth Fong, and Margo Seltzer. People of data: The end-to-end provenance project. *Patterns*, 1, May 2020.

[29] Patrick Esser et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Proc. of ICML*, 2024.

[30] Hugging Face. Hugging face models. https://huggingface.co/models, 2025.

[31] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proc. of ICCV*, 2023.

[32] FFusion. Base 768 alpha ffusion.ai. https://huggingface.co/FFusion/di.FFUSION.ai-v2.1-768-BaSE-alpha, 2023.

[33] Nielsen Frank. On a generalization of the jensen-shannon divergence and the js-symmetrization of distances relying on abstract means. *arXiv preprint arXiv:1904.04017*, 2019.

[34] Peng Gao et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024.

[35] Aaron Gokaslan et al. CommonCanvas: Open diffusion models trained on creative-commons images. In *Proc. of CVPR*, pages 8250–8260, 2024.

[36] Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[37] Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Trap: Targeted random adversarial prompt honeypot for black-box identification. *arXiv preprint arXiv:2402.12991*, 2024.

[38] Ji Guo, Wenbo Jiang, Rui Zhang, Guoming Lu, and Hongwei Li. One prompt to verify your models: Black-box text-to-image models verification via non-transferable adversarial attacks. *arXiv preprint arXiv:2410.22725*, 2024.

[39] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[40] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: towards a better evaluation metric for image generation. In *Proc. of CVPR*, 2024.

[41] Hengrui Jia, Hongyu Chen, Jonas Guan, Ali Shahin Shamsabadi, and Nicolas Papernot. A zest of lime: Towards architecture-independent model distances. In *Proc. of ICLR*, 2021.

[42] Heng Jin, Chaoyu Zhang, Shanghao Shi, Wenjing Lou, and Y Thomas Hou. Proflingo: A fingerprinting-based intellectual property protection scheme for large language models. In *Proc. of CNS*, 2024.

[43] Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding ddpm latent codes through optimal transport. *arXiv preprint arXiv:2202.07477*, 2022.

[44] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proc. of ICML*, 2023.

[45] Kohya S. SD Scripts. https://github.com/kohya-ss/sd-scripts, 2024.

[46] Minoru Kuribayashi, Tatsuya Yasui, and Asad Malik. White box watermarking for convolution layers in fine-tuning model using the constant weight code. *Journal of Imaging*, 2023.

[47] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Proc. of NeurIPS*, 2023.

[48] Yanyu Li et al. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Proc. of NeurIPS*, 2023.

[49] Yuanchun Li, Ziqi Zhang, Bingyan Liu, Ziyue Yang, and Yunxin Liu. Modeldiff: Testing-based dnn similarity comparison for model reuse detection. In *Proc. of ACM ISSTA*, 2021.

[50] Zhimin Li et al. Hunyuan-DiT: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024.

[51] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 1991.

[52] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Proc. of NeurIPS*, 36:34892–34916, 2023.

[53] Shayne Longpre et al. Data authenticity, consent, and provenance for ai are all broken: What will it take to fix them? *MIT Media Lab Reports*, 2024.

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

[55] Lummi.ai. https://www.lummi.ai.

[56] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Proc. of NeurIPS*, 2017.

[57] Peizhuo Lv et al. A robustness-assured white-box watermark in neural networks. *IEEE Trans. on DSC*, 2023.

[58] Zhiyuan Ma et al. Efficient diffusion models: A comprehensive survey from principles to practices. *IEEE TPAMI*, 2025.

[59] Midjourney. Midjourney. https://www.midjourney.com/home, 2024.

[60] Xin Mu, Yu Wang, Yehong Zhang, Jiaqi Zhang, Hui Wang, Yang Xiang, and Yue Yu. Model provenance via model dna. *arXiv preprint arXiv:2308.02121*, 2023.

[61] Nilaier. Waifu-diffusers. https://huggingface.co/Nilaier/Waifu-Diffusers, 2023.

[62] Nutbutter. bigasp v2.0. https://civitai.com/models/502468/bigasp, 2024.

[63] Bofeng Pan, Natalia Stakhanova, and Suprio Ray. Data provenance in security and privacy. *ACM Computing Surveys*, 55(14s):1–35, 2023.

[64] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proc. of CVPR*, 2022.

[65] Dustin Podell et al. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[66] PseudoTerminal X. Photo Concept Bucket. https://huggingface.co/datasets/bghira/photo-concept-bucket, 2024.

[67] Alec Radford et al. Learning transferable visual models from natural language supervision. In *Proc. of ICML*, 2021.

[68] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" explaining the predictions of any classifier. In *Proc. of KDD*, 2016.

[69] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. of CVPR*, 2022.

[70] Nataniel Ruiz et al. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. of CVPR*, 2023.

[71] Scenario.gg. AI-generated game assets. https://www.scenario.gg/, 2022.

[72] Christoph Schuhmann. LAION-Aesthetics. https://laion.ai/blog/laion-aesthetics/, 2022.

[73] Christoph Schuhmann et al. LAION-5B: An open large-scale dataset for training next generation image-text models. 2022.

[74] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When do we not need larger vision models? In *Proc. of ECCV*, 2024.

[75] James Seale Smith et al. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *Transactions on Machine Learning Research*, 2023.

[76] Wei Song, Wen Ma, Ming Zhang, Yanghao Zhang, and Xiaobing Zhao. Lightweight diffusion models: a survey. *Artificial Intelligence Review*, 2024.

[77] Stability AI. Stable diffusion 2.1 release. https://stability.ai/news/stablediffusion2-1-release7-dec-2022, December 2022.

[78] Stability AI. Introducing stable diffusion 3.5. https://stability.ai/news/introducing-stable-diffusion-3-5, October 2024.

[79] Stability AI. Stable Diffusion v1-5 model card. https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5, 2024.

[80] Stability AI. Stable Diffusion v2-1 model card. https://huggingface.co/stabilityai/stable-diffusion-2-1, 2024.

[81] Beth Stackpole. Bringing transparency to the data used to train artificial intelligence. https://mitsloan.mit.edu/ideas-made-to-matter/bringing-transparency-to-data-used-to-train-artificial-intelligence, 2025.

[82] strangerzonehf. Flux-super-realism-lora. https://huggingface.co/strangerzonehf/Flux-Super-Realism-LoRA, 2024.

[83] Yuchen Sun et al. Deep intellectual property protection: A survey. *arXiv preprint arXiv:2304.14613*, 2023.

[84] Kolors Team. Kolors: Effective training of diffusion model for photorealistic text-to-image synthesis. *arXiv preprint*, 2024.

[85] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[86] Tess. https://www.tess.design.

[87] Jim Thacker. Magma adds new 'ethically responsible' ai art tools. https://www.cgchannel.com/2023/06/magma-adds-new-ethically-responsible-generative-ai-tools/, 2023.

[88] Tony Ho Tran. Image Apps Like Lensa AI Are Sweeping the Internet, and Stealing From Artists. https://www.thedailybeast.com/how-lensa-ai-and-image-generators-steal-from-artists, 2022.

[89] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proc. of AAAI*, 2023.

[90] Tianhao Wang and Florian Kerschbaum. Riga: Covert and robust white-box watermarking of deep neural networks. In *Proc. of WWW*, 2021.

[91] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Proc. of NeurIPS*, 2024.

[92] Enze Xie et al. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proc. of ICCV*, 2023.

[93] Enze Xie et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[94] Ziqing Yang, Yixin Wu, Yun Shen, Wei Dai, Michael Backes, and Yang Zhang. The challenge of identifying the origin of black-box large language models. *arXiv preprint arXiv:2503.04332*, 2025.

[95] Yiren Lu. Fine-tuning a FLUX.1-dev style LoRA. https://modal.com/blog/fine-tuning-flux-style-lora, 2024.

[96] Zhuomeng Zhang, Fangqi Li, Chong Di, and Shilin Wang. PromptLA: Towards integrity verification of black-box text-to-image diffusion models. *arXiv preprint arXiv:2412.16257*, 2024.

[97] Wendi Zheng et al. Cogview3: Finer and faster text-to-image generation via relay diffusion. In *Proc. of ECCV*, 2024.

A Appendix

A.1 Proof of Theorem 1

We provide the proof for Theorem 1 that justifies our choice of measuring distance between models from their outputs.

Proof. Given two models $\mathcal{M}_A$ and $\mathcal{M}_B$, to show that $J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p)) = J(O(\mathcal{M}_A|p), O(\mathcal{M}_B|p))$, it is essential to show $I(\mathcal{M}_*|p) = O(\mathcal{M}_*|p)$ for $\mathcal{M}_A$ and $\mathcal{M}_B$. Without loss of generality, we demonstrate the result for $\mathcal{M}_A$ and the conclusions apply to $\mathcal{M}_B$ under the same assumptions.

We assume that $\mathcal{M}_A$ is a perfectly trained diffusion model and its training data distribution on prompt p follows the Gaussian distribution $N(\mu, \sigma^2)$.

At run-time, given prompt p and a noise seed ε sampled from $N(0,1)$, $\mathcal{M}_A$ will generate an output defined by $\sigma \cdot \varepsilon + \mu$. This result directly follows Theorem 3.1 in [43]. The proof in [43] shows that for a perfectly-trained diffusion model, the diffusion process is the same as the Monge optimal transport map between $N(0,1)$ and $N(\mu, \sigma^2)$.

Therefore, under our assumptions, $I(\mathcal{M}_A|p) = O(\mathcal{M}_A|p)$ for the perfectly-trained diffusion model $\mathcal{M}_A$. Similarly, for a perfectly-trained $\mathcal{M}_B$, the output distribution is the same as the input distribution, $I(\mathcal{M}_B|p) = O(\mathcal{M}_B|p)$. Therefore, we prove that $J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p)) = J(O(\mathcal{M}_A|p), O(\mathcal{M}_B|p))$. □

A.2 Proof of Corollary 1

We prove the corollary that the distance between two base models decreases with the variance of their input distributions.

Proof. We follow the assumptions that two base models $\mathcal{M}_A$ and $\mathcal{M}_B$ are perfectly trained diffusion models. In addition, the input distributions of the models given prompt p are Gaussian. We assume that $I(\mathcal{M}_A|p)$ and $I(\mathcal{M}_B|p)$ have the same variance σ^2. Specifically, $I(\mathcal{M}_A|p)$ has Gaussian distribution $N(\mu_A, \sigma^2)$ and $I(\mathcal{M}_B|p)$ has Gaussian distribution $N(\mu_B, \sigma^2)$.

By Equation 1, the distance between $\mathcal{M}_A$ and $\mathcal{M}_B$ is measured from the Jensen-Shannon divergence between their output distributions $O(\mathcal{M}_A|p)$ and $O(\mathcal{M}_B|p)$ using prompt p. By Theorem 1, we have $J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p)) = J(O(\mathcal{M}_A|p), O(\mathcal{M}_B|p))$. Therefore, we will prove that $J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p))$ decreases with σ.

The Jensen-Shannon divergence between two Gaussian distributions $N(\mu_A, \sigma_A^2)$ and $N(\mu_B, \sigma_B^2)$ is approximated by :

$$\frac{1}{2}\left(\frac{\mu_A^2}{2\sigma_A^2} + \frac{\mu_B^2}{2\sigma_B^2} - \frac{\mu_*^2}{\sigma_*^2} + \log_2 \frac{\sigma_A^2 + \sigma_B^2}{2\sigma_A\sigma_B}\right) \tag{6}$$

where $\sigma_*^2 = \frac{2\sigma_A^2\sigma_B^2}{\sigma_A^2+\sigma_B^2}$ and $\mu_* = \sigma_*^2(\frac{\mu_A}{2\sigma_A^2} + \frac{\mu_B}{2\sigma_B^2})$. Equation 6 is a direct application of Corollary 10 from [33].

Applying Equation 6 to the distributions of $I(\mathcal{M}_A|p)$ and $I(\mathcal{M}_B|p)$, we have

$$J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p)) = \frac{(\mu_A - \mu_B)^2}{8\sigma^2} \tag{7}$$

Therefore, the distance between $\mathcal{M}_A$ and $\mathcal{M}_B$ is $\frac{(\mu_A-\mu_B)^2}{8\sigma^2}$ which decreases with the variance σ^2.

□

A.3 Proof of Theorem 2

We now provide proof for Theorem 2 that demonstrates effectiveness of our model provenance method.

Proof. To quantify the distance between two models $Dist(\mathcal{M}_A^F, \mathcal{M}_*|p)$, we leverage Theorem 1 and Equation 1 to quantify the divergence between models' input distributions $J(I(\mathcal{M}_A^F|p), I(\mathcal{M}_*|p))$.

We assume that two base models, $\mathcal{M}_A$ and $\mathcal{M}_B$, generate outputs when given prompt p from Gaussian distribution $N(\mu_A, \sigma_A^2)$ and $N(\mu_B, \sigma_B^2)$ respectively. Two base models are separable under prompt p and their distance satisfies $Dist(\mathcal{M}_A, \mathcal{M}_B|p) \gg 0$ because their input distributions are very difference, *i.e.,* $J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p)) \gg 0$.

Let $\mathcal{M}_A^F$ be a fine-tuned model from $\mathcal{M}_A$ and the fine-tuning data is much smaller in size compared to the original training data of $\mathcal{M}_A$. As $\mathcal{M}_A^F$ is fine-tuned on $\mathcal{M}_A$, its training data consists of training data of $\mathcal{M}_A$ and the fine-tuning data. Suppose the fine-tuning data of p has Gaussian distribution $N(\mu_F, \sigma_F^2)$, then the overall training data of p for $\mathcal{M}_A^F$ has Gaussian mixture distribution $(1-w)\cdot N(\mu_A, \sigma_A) + w \cdot N(\mu_F, \sigma_F)$ for some $w \in [0,1]$. Clearly, $w = 0$ if the semantic space of p is not affected by the fine-tuning. Therefore, the probability of $w = 0$ is the portion of the semantic space that is affected by the fine-tuning. Because the fine-tuning data is much smaller in size compared to the training data of $\mathcal{M}_A$, very little of the semantic space is modified by the fine-tuning, thus we assume the weight coefficient $w = 0$ with a very high probability, for most test prompts. Since the provenance system selects multiple semantically unrelated test prompts and applies majority vote to minimize the impact of fine-tuning, we hereby assume $w \to 0$ for each test prompt p.

With $w \to 0$, we can simplify the Gaussian mixture distribution through moment matching. We assume the overall training distribution of $\mathcal{M}_A^F$ can be simplified by a single Gaussian distribution with mean $\mu_{F'} = (1-w)\cdot\mu_A + w\cdot\mu_F$ and variance $\sigma_{F'}^2 = (1-w)\cdot(\mu_A^2 + \sigma_A^2) + w\cdot(\mu_F^2 + \sigma_F^2) - ((1-w)\cdot\mu_A + w\cdot\mu_F)^2$. This approximation is the best as of Kullback-Leibler divergence in the exponential family of distributions [6].

We can now compute the distance between models using the Jensen-Shannon Divergence J. As $w \to 0$, we have $\lim_{w\to 0}\mu_{F'} = \mu_A$ and $\lim_{w\to 0}\sigma_{F'}^2 = \sigma_A^2$. Therefore, the prompt-specific model distance between $\mathcal{M}_t$ and $\mathcal{M}_A$ satisfies

$$\lim_{w\to 0} J(I(\mathcal{M}_A^F|p), I(\mathcal{M}_A|p)) = 0.$$

On the other hand, the distance between $\mathcal{M}_A^F$ and $\mathcal{M}_B$ is

$$\lim_{w\to 0} J(I(\mathcal{M}_A^F|p), I(\mathcal{M}_B|p)) = J(I(\mathcal{M}_A|p), I(\mathcal{M}_B|p))$$

Thus we have $Dist(\mathcal{M}_A^F, \mathcal{M}_A|p) \approx 0$ and $Dist(\mathcal{M}_{A^F}, \mathcal{M}_B|p) \approx Dist(\mathcal{M}_A, \mathcal{M}_B|p) \gg 0$.

Therefore, by comparing the model distances, we can effectively attribute the fine-tuned model $\mathcal{M}_A^F$ to its base $\mathcal{M}_A$ using our provenance method. □

A.4 Detailed prompts

We list some samples of the provenance test prompts.

- "`Bassinet`": A realistic photo of a baby bassinet placed in a cozy nursery. The bassinet is made of light wood and has soft white bedding and a sheer canopy. The nursery features warm lighting, a plush rug, pastel-colored walls, and gentle decorations like stuffed animals and framed baby artwork. The scene is clean, serene, and welcoming.
- "`Cake`": A beautifully decorated cake placed on a table. The cake is multi-layered with intricate floral designs in pastel colors like pink, blue, and white. It has elegant frosting details and is surrounded by soft lighting for an inviting look. The table is set with a few matching plates and utensils, giving a warm and festive atmosphere.
- "`Camp`": Peaceful forest camp in daylight. A dome tent sits on grassy clearing, surrounded by tall pine trees. Stone fire ring with small fire and log seats in front. Camping gear and a lantern beside the tent. Sunlight filters through trees, casting warm dappled shadows. A dirt trail leads into the forest. Cozy, quiet, natural outdoor setting.
- "`Blazer`": A stylish modern gray blazer displayed on a neutral mannequin in a well-lit studio. The blazer features clean, sharp tailoring, textured high-quality fabric, and a minimalist, elegant design. Soft, even lighting highlights its structure and craftsmanship. A simple background with gentle shadows draws attention to the garment's details, keeping the focus on the blazer.

A.5 Fine-tuning Setup

For fine-tuning SD2.1, we use the default learning rate of $1e-4$ to vary the dataset size (100k - 500k images) and the number of fine-tuning epochs. We also tested two learning rates ($1e-5$ and $1e-6$) on 300k images. To fine-tune FLUX, we use a recommended learning rate of $1e-6$ [1,95], as higher learning rates (e.g., $1e-4$) produce poor image quality. This $1e-6$ setting ensures a stable and visually clean output.

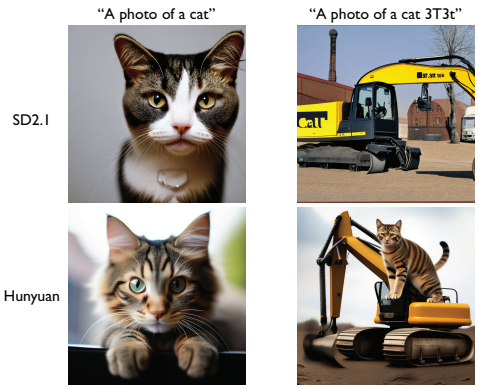


Figure 10: Although it was intended only for SD2.1, the adversarial prompt ('3T3t') [38] causes a similar foreign artifact in Hunyuan, making it ineffective for our problem setting.

Figure 11: Samples of images generated from short, simple prompts ("`photo of a cake`") show large variations in shape, color, background, and details, making provenance detection harder. Using detailed prompts, as in Figure 2, gives more consistent outputs that better reveal model-specific characteristics.

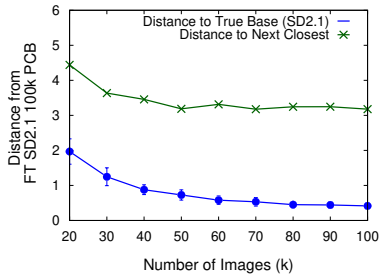


Figure 12: Impact of sampling parameter k on provenance performance. We measure the distance from a fine-tuned SD2.1 model (using 100K PCB) to all base models. The distance to its true base (SD2.1) decreases and converges with k, but remains much lower than the distance to the next closest base model. At $k = 100$, the system achieves stable separation with high confidence.

Model	**Model Distance from SD2.1**
Ft-SD2.1-100k	0.42 ± 0.09
Ft-SD2.1-200k	0.62 ± 0.13
Ft-SD2.1-300k	0.45 ± 0.09
Ft-SD2.1-400k	0.38 ± 0.09
Ft-SD2.1-500k	0.53 ± 0.13
Ft-SD2.1-300k-4epoch	0.40 ± 0.07
Ft-SD2.1-300k-8epoch	0.36 ± 0.06
Ft-SD2.1-300k-1e-5	0.38 ± 0.07
Ft-SD2.1-300k-1e-6	0.33 ± 0.05
DeepFloyd	3.11 ± 0.41
SDXL	2.74 ± 0.43
Kolors	3.47 ± 0.44
Pixart-Σ	7.03 ± 1.04
Hunyuan-DiT	3.16 ± 0.50
FLUX	4.22 ± 0.72
Sana	5.95 ± 0.86

Table 4: Model distance from SD2.1 to fine-tuned variants of SD2.1 using the PCB dataset and to other base models, using the detailed prompt on "sofa."

Base Model	**SD2.1 with 100K PCB ("sofa")**	
	S2-CLIP	**CLIP**
SD2.1	**0.42 ± 0.09**	**0.76 ± 0.12**
DeepFloyd	3.68 ± 0.64	4.95 ± 0.87
SDXL	3.34 ± 0.57	5.71 ± 1.14
Kolors	4.13 ± 0.44	11.66 ± 2.48
PixArt-Σ	6.25 ± 1.09	7.14 ± 1.50
Hunyuan	(3.18 ± 0.59)	5.79 ± 0.91
FLUX	4.15 ± 0.68	(3.93 ± 0.51)
Sana	6.18 ± 0.85	11.88 ± 2.36

Table 5: Comparison of model distances measured using S2-CLIP and CLIP for provenance detection. S2-CLIP achieves a larger separation between the true base model and the next closest candidate, improving detection reliability. We mark the smallest distance by bold, and the second smallest by ( ).