FedOT: Ownership Verification and Leakage Tracing via Watermarks for Federated LDMs

Wenlong Cheng[1], Yuan Gan[2*], Yunqiu Xu[3], and Jiaxu Miao[4*]

[1]Northwest Normal University, China
[2]The University of Tokyo, Japan
[3]National University of Singapore, Singapore
[4]Sun Yat-sen University, China
2023222209@nwnu.edu.cn, y-gan@mi.t.u-tokyo.ac.jp,
imyunqiuxu@gmail.com, miaojx@mail.sysu.edu.cn

Abstract. Training Latent Diffusion Models (LDMs) within Federated Learning (FL) has attracted increasing attention due to its ability to combine the powerful generative capacity of LDMs with the privacy-preserving properties of FL. However, FL requires sharing the global model with multiple participants, which risks unauthorized model distribution or resale by malicious clients. While an intuitive approach is to adopt existing VAE-based watermarking techniques for LDMs in FL, this strategy falls short in addressing such threats due to two fundamental challenges: (1) Existing methods support ownership verification but lack the ability to trace model leakage to a specific malicious client; (2) VAE-based watermarks are vulnerable, as they can be removed simply by replacing the decoder with a clean counterpart. In this paper, we propose FedOT, the first framework for ownership verification and leakage tracing in federated LDMs. Specifically, to address the first challenge, we design a chunked watermark, where the first part is for ownership verification, and the second part is used for client identification. Furthermore, to overcome the second challenge and secure the model against VAE replacement attack, we introduce Latent Vector Transformation (LVT), which strengthens the connection between the VAE and U-Net latent spaces by modifying the original latent distribution of the VAE. Consequently, any attempt to replace the VAE for watermark removal leads to significant image quality degradation, making the LDM model unusable. Extensive experiments demonstrate that FedOT achieves superior performance in both ownership verification and traceability. Project page: https://spyzixuan.github.io/FedOT/.

Keywords: Federated Learning · Latent Diffusion Models · Models Ownership Verification · Models Source Tracing · Model Watermark

1 Introduction

Latent Diffusion Models (LDMs) [1,12,31,33,36] have demonstrated remarkable capabilities in high-fidelity image synthesis and reshaped the AIGC landscape [4,

* Corresponding authors.

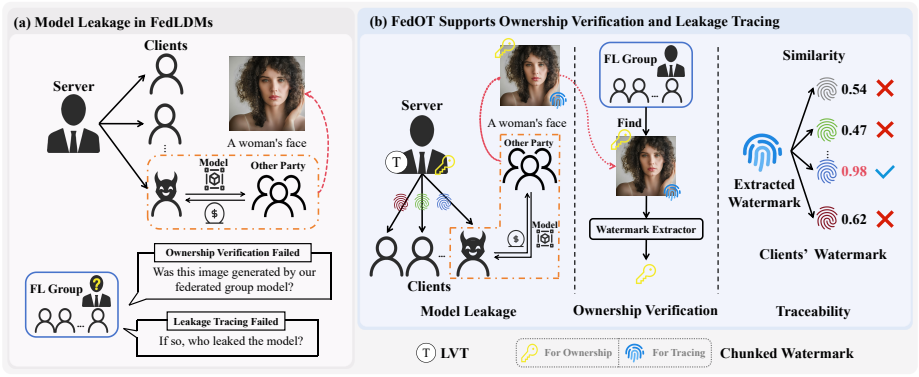

Fig. 1: Motivation of FedOT. (a) Malicious clients leak the federated LDMs, verifying ownership or tracing the source of generated images becomes infeasible. (b) **FedOT supports ownership verification and malicious client tracing even if the leaked model is misused**, by using chunked watermark and Latent Vector Transformation (**LVT**).

52]. Traditionally, these models rely on centralized training over massive open-source datasets. To achieve privacy-preserving model training, federated Latent Diffusion Models (FedLDMs) [11, 21, 24, 29, 42] synergize the strengths of LDMs with the privacy guarantees of Federated Learning (FL) [27], enabling diffusion models to learn from distributed clients while preserving data privacy.

Despite successfully safeguarding data privacy, FedLDMs introduce a critical vulnerability concerning model ownership. As illustrated in Fig. 1(a), the sharing of the global model among participants inevitably exposes the intellectual property to malicious clients. They may distribute or resell the fine-tuned models without authorization. Such leakage of models not only constitutes intellectual property infringement but also raises ethical concerns, including harmful content generation and privacy leakage [3, 32].

While recent works [39, 43] have explored copyright protection and traceability in FL, these solutions are exclusively designed for classification tasks. The substantial architectural differences render them fundamentally inapplicable to complex generative models like LDMs. To address this gap, we must look toward watermarking techniques specifically designed for LDMs [9, 17, 19, 28, 46, 47, 53]. However, adapting these techniques to FL requires careful consideration of the training process. During LDM fine-tuning, clients typically only update the U-Net parameters, leaving the VAE entirely frozen. Given this training characteristic, the VAE-based watermarking approach [9] emerges as a highly intuitive and promising candidate for FL. Despite its theoretical suitability, directly deploying this technique in a federated setting exposes two fundamental challenges: ❶ **It can only verify that a model originates from the FL group (*i.e.*, ownership verification), but cannot identify the specific malicious client responsible for leaking the model (*i.e.*, leakage tracing)**; ❷ **The watermark can be easily removed without degrading the model's utility by replacing the watermark VAE decoder with a clean counterpart**.

To address these limitations, this paper proposes **FedOT, the first framework capable of both ownership verification and leakage tracing for FedLDMs**. Rather than focusing on watermarking algorithms, our core design introduces a chunked watermark mechanism. Specifically, before distributing the global model, the server embeds different binary watermarks into the VAE decoder of each client's LDM. The first r bits identify whether the model originates from the federated group. The remaining $n-r$ bits are unique to each client and are used to trace the source of the leak. If a malicious client leaks the model, we can extract the watermark from the images generated by the leaked model. An advantage of this design is that it runs the full tracing process only when the first r bits confirm the model's origin, making detection more efficient.

However, simply embedding the watermark into the VAE is vulnerable. Malicious clients can effortlessly execute a zero-cost replacement attack by swapping the watermarked VAE with an available clean counterpart. This effectively removes the watermark without degrading the quality of the generated images. To prevent such replacement attacks, we propose Latent Vector Transformation (LVT), a technique designed to tightly bind the VAE and U-Net components. Rather than modifying the watermarking algorithm, LVT proactively alters the latent space distribution of the VAE. Consequently, the U-Net is forced to gradually adapt to this modified distribution during the federated training process.

To balance the binding strength and generation quality, we explore three types of LVT strategies, namely translation, mirror, and negative, which modify the latent distribution to secure our framework against replacement attacks.

We systematically analyze the distribution shift of the three LVT strategies. Our experiments demonstrate that the negative transformation is the optimal choice considering the performance and component binding strength. Overall, our main contributions can be summarized as follows:

- To the best of our knowledge, we propose the first framework, FedOT, that enables both ownership verification and leakage tracing for FedLDMs, filling a critical gap in existing research.
- We introduce a VAE Latent Vector Transformation, ensuring that the VAE remains compatible only with the U-Net trained in FL. Our proposed LVT establishes a strong dependency between model components: any attempt to remove the watermark by replacing the VAE incurs severe degradation in image quality, thereby deterring malicious clients from doing so.
- We conduct experiments under various common watermark removal attacks, such as VAE replacement attack, model purification attack, and image attacks. The results show that FedOT provides a reliable mechanism for ownership verification and leakage tracing for FedLDMs.

2 Related Work

Federated Learning (FL). FL enables multiple clients to collaboratively train models without sharing raw data, thereby preserving privacy. In a typical client-server architecture [8, 50], each client updates the model with local data, while

the server aggregates these updates using FedAvg [27] to form a global model. We adopt FL to train Latent Diffusion Models [36] for decentralized generative modeling, where only the U-Net parameters are uploaded and aggregated each round, reducing communication costs. However, FL also introduces new challenges for copyright protection, as the global model is accessible to all participating clients.

Diffusion Models. Denoising Diffusion Probabilistic Models (DDPMs) [15] generate data through iterative noise addition and removal. Latent Diffusion Models (LDMs) [36] improve efficiency by performing this process in a low-dimensional latent space, where images are encoded, denoised, and reconstructed by a decoder, or synthesized from a Gaussian prior via DDIM [41]. Diffusion models have achieved remarkable success in text-to-image generation [1,12,16,35,56] and editing tasks [2, 18, 37, 49, 55]. Stable Diffusion [7, 33, 36], a widely adopted open-source implementation, has further driven research and applications. Despite their strong generative capabilities, LDMs typically rely on centralized training with large-scale data, raising significant privacy concerns [10, 11, 24, 44].

Digital Watermark. Watermarking is crucial for model traceability and intellectual property protection, spanning data watermarking [54], image watermarking [6, 23, 57], and model watermarking [5, 9, 17, 26, 28, 46–48, 53]. For LDM-based generative models, watermarks can be embedded in generated images, model parameters, or latent space, each with distinct tradeoffs in robustness and flexibility. In FL, server-side methods are preferred over client-side approaches [20, 25, 51] to prevent malicious backdoor injection. WAFFLE [43] pioneered server-side FL watermarking for ownership verification, while FedTracker [39] further enables per-client traceability but is limited to classification models. We propose FedOT to fill the gap in watermark tracing and ownership verification for generative models in FL.

3 Methodology

3.1 FedOT Framework

As shown in Fig. 2(a), our FedOT framework consists of two main components: Latent Vector Transformation (LVT) and watermark embedding. FedOT is designed to collaboratively fine-tune Stable Diffusion [36] on private distributed data in a federated setting. To enhance intellectual property protection, the trusted server in FedOT first performs LVT training on the VAE module of the initial global model M, enabling the VAE to learn a transformed latent distribution and to produce a globally consistent latent representation. Based on this adapted global model M_T, the server then creates K model replicas $\{M_i\}_{i=1}^{K}$, each injected with a unique n-bit watermark, resulting in the watermarked models $\{\hat{M}_i\}_{i=1}^{K}$. These watermarked models are then distributed to clients for local fine-tuning on their private data. After each training round, clients upload their updated U-Net parameters and a subset of generated samples, which are aggregated on the server and redistributed for the next round of training. A watermark extractor from the VAE-based method [9] retrieves watermarks from the generated images. The overall FedOT procedure is illustrated in Algorithm 1.

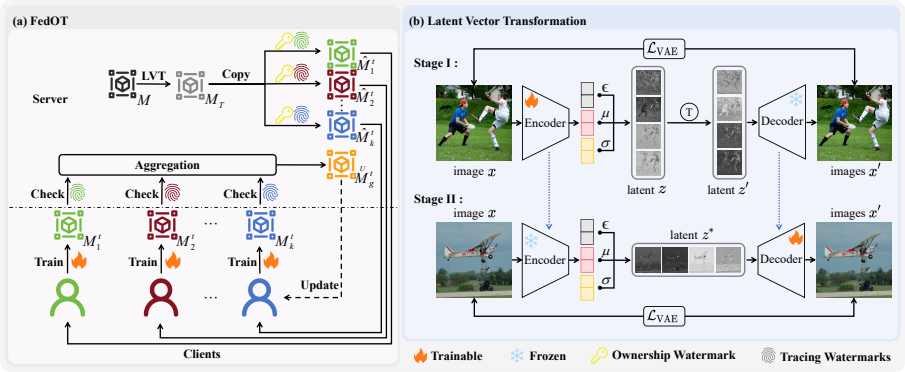

Fig. 2: Overview of FedOT. (a) **FedOT Workflow.** The server applies LVT to the VAE, then embeds watermarks, and distributes model replicas to clients. Clients train the SD model locally and upload updates. The server verifies watermarks and aggregates only U-Net parameters. (b) **Details of LVT within FedOT.** Stage I: Transform latent vector z to z' and train the encoder. Stage II: Use the trained encoder to generate z^* and adapt the decoder accordingly.

If a malicious client leaks or resells the local LDM, the embedded watermark in generated images can be extracted to verify ownership and trace the source. The FL group first confirms ownership, then identifies the leaking client. More background details are provided in Appendix A.

3.2 Watermark Design and Training

Watermark Design. We propose a chunked watermark that supports both ownership verification and tracing. Specifically, the n-bit watermark is divided into two parts: the first r bits for ownership verification, and the remaining $n-r$ bits for client tracing. If the extracted watermark $\mathbf{m}'$ matches the originally embedded watermark $\mathbf{m}$ under the following condition, the image is verified as originating from the federated model:

$$\text{Verify}(\mathbf{m}', \mathbf{m}) = \begin{cases} \text{True}, & \text{Match}(\mathbf{m}'_{1:r}, \mathbf{m}_{1:r}) \geq \tau \\ \text{False}, & \text{otherwise}, \end{cases} \tag{1}$$

where $\text{Match}(\cdot)$ denotes the bit accuracy between two binary watermarks (Appendix B.3). Upon successful verification, the server proceeds to trace the specific client responsible for the leak:

$$j = \text{argmax } \text{Trace}(\mathbf{m}'_{r+1:n}, \mathbf{m}_{i,r+1:n}), \tag{2}$$

where j is the client index, indicating the j-th client leaked the model. This chunked watermark first verifies ownership via the shared r-bit prefix, and only performs full client identification using the $n-r$ suffix when necessary, reducing overhead in large-scale deployments.

Watermark Training. During LDMs training, only the U-Net is updated while the VAE is frozen [13,37]. To ensure the embedded watermark remains unaffected

Algorithm 1 Training Pipeline of FedOT

Input: Global model M, number of clients K, dataset $\{D_i\}_{i=1}^{K}$, watermark length n, verification length r, transformation type T, training steps S
Output: Final aggregated global model M_g^T

Train global model's VAE to learn the LVT transformation
$E_T, D_T \leftarrow LVTTraining(M, T)$ ▷ Appendix C, Algorithm 4
Update the global model's VAE
$M_T \leftarrow UpdateGlobalModel(M, E_T, D_T)$
Generate replicas with unique watermarks after LVT
$\{\hat{M}_i\}_{i=1}^{K} \leftarrow WatermarkEmbedding(M_T, K, n, r)$ ▷ Appendix B, Algorithm 3
Perform federated training using watermarked model replicas
$M_g^T \leftarrow FederatedTrainingSD(\{\hat{M}_i\}_{i=1}^{K}, \{D_i\}_{i=1}^{K}, S)$ ▷ Appendix A, Algorithm 2
return M_g^T

during FL, inspired by Stable Signature [9], we embed the watermark into the decoder of the VAE. Before FL training, the trusted server embeds a watermark into the VAE decoder for each client. This watermark training is performed only once per client and does not need to be repeated in each communication round. To embed a unique watermark for each client, the server adopts the watermark extractor from Stable Signature to guide the training of the VAE decoder. Specifically, we use a publicly available dataset [22] to generate latent vectors z via a VAE encoder. The decoder then reconstructs the image $\hat{x}$ from z, and $\hat{x}$ is input to the watermark extractor to predict the embedded watermark $\mathbf{m}'$. The Binary Cross-Entropy (BCE) loss is applied between $\mathbf{m}'$ and the target watermark $\mathbf{m}$ to ensure successful embedding:

$$\mathcal{L}_m = -\sum_{i=1}^{k} \left[\mathbf{m}_i \cdot \log \sigma(\mathbf{m}'_i) + (1 - \mathbf{m}_i) \cdot \log\left(1 - \sigma(\mathbf{m}'_i)\right)\right]. \tag{3}$$

To preserve image quality, we additionally apply the Watson-VGG loss to encourage $\hat{x}$ to remain perceptually similar to the original input image x:

$$\mathcal{L}_i = \text{Watson-VGG}\left(\hat{x}, x\right), \tag{4}$$

where a weighted coefficient λ_i is used to balance image fidelity and watermark bit accuracy, and the final objective can be formulated as:

$$\mathcal{L}_w = \mathcal{L}_m + \lambda_i \cdot \mathcal{L}_i. \tag{5}$$

Fine-tuning the VAE decoder with this loss keeps the watermark intact despite U-Net updates, avoiding training interference throughout the entire optimization process. More training details are provided in Appendix B.

3.3 Latent Vector Transformation

Since the VAE is typically frozen during LDM fine-tuning, malicious clients can easily replace the watermarked VAE with an open-source version. This effec-

tively erases the embedded watermarks without incurring any degradation in image generation quality. To prevent this, we propose Latent Vector Transformation (LVT) to establish a strong dependency between the VAE and U-Net in the latent space. Our key insight relies on the training mechanism of LDMs: the U-Net gradually adapts to the latent distribution produced by the VAE encoder. Consequently, modifying this distribution forces the U-Net to shift toward a new latent space, subsequently causing it to "forget" the original pre-trained distribution. Exploiting this property, we fine-tune the global VAE to transform its latent space prior to federated training. This process effectively binds the subsequently trained U-Net to our modified VAE, rendering any decoder replacement attacks destructive to the model's utility.

LVT Training. Before watermark embedding, the server fine-tunes the VAE using a public dataset [22] to modify the latent distribution in a controlled manner. We introduce a transformation T to adjust latent vectors, ensuring the U-Net operates only on the transformed space. As shown in Fig. 2(b), the LVT process includes two stages. In stage I, the encoder is trained to learn the transformation T, while the decoder remains fixed. Given an image x, the VAE encoder maps it to a latent vector $z = E(x)$, which is then transformed into $z' = T(z)$. The decoder reconstructs the final image as $x' = D(z')$. To faithfully reconstruct the original image from z', the encoder implicitly learns to approximate the transformation T during training. In stage II, the encoder is kept frozen, and the decoder is fine-tuned. Since the encoder has already learned the transformation T during stage I, its current output z^* differs from the original latent vector z. The decoder is trained to reconstruct images from the transformed latent space z^* by implicitly learning the inverse transformation T^{-1}, allowing the decoder to adapt to the new latent space.

However, changes in the latent space can pose challenges for FL, as the U-Net needs to adapt to the new latent distribution. Therefore, it is critical to design LVT strategies with an optimal balance: they must induce a sufficient structural shift to break compatibility with clean VAEs, while minimizing any adverse impact on the final generative fidelity of the U-Net.

Random Transformation. The latent vector z is typically sampled via the reparameterization trick from the encoder output mean μ and variance σ, formulated as $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Our objective is to fine-tune the VAE so that the latent space adopts a new structural transformation while preserving its Gaussian properties.

First, based on the properties of the Gaussian distribution, we consider adding a random normal distribution $\hat{\epsilon}$ to the latent vector z, yielding $z' = z + \hat{\epsilon}$. Under this transformation, z' still follows a Gaussian distribution:

$$z' \sim \mathcal{N}(\mu, I(\sigma^2 + 1)). \tag{6}$$

However, results reveal that applying random Gaussian transformations significantly degrades the image reconstruction quality. We denote this method as $\mathbf{FedOT_{rand}}$ and include it as a baseline for comparison. Our analysis indicates that the VAE has difficulty adapting random Gaussian transformations.

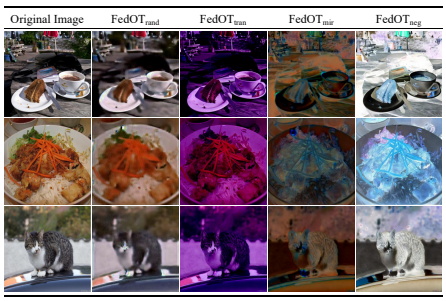

Fig. 3: VAE reconstruction results after training the encoder with different LVT transformations.

Although standard Gaussian noise is added in each training iteration, the VAE cannot learn a consistent deterministic mapping to counteract this perturbation. To seek a deterministic approach, we draw inspiration from recent findings [30], which preliminarily observed that applying constant shifts to the latent space can predictably alter generative attributes like color. We significantly advance this insight: rather than merely manipulating visual outputs, we formalize systematic, non-stochastic latent shifts as a mechanism to structurally bind LDM components against adversarial replacement attacks. Building on this novel perspective, we design and explore the following three stable LVT strategies.

Translation Transformation. Leveraging the linearity of Gaussian distributions, we apply a deterministic translation to the latent vector z, defined as $z' = z + c$. Under this operation, the transformed latent variable z' follows:

$$z' \sim \mathcal{N}(\mu + c, \sigma^2). \tag{7}$$

We denote this method as $\mathbf{FedOT_{tran}}$. This strategy shifts the distribution mean while preserving the original variance structure. During Stage I fine-tuning, the VAE encoder learns the translation T, effectively mapping inputs to a globally shifted latent space. Subsequently, in Stage II, the frozen encoder produces shifted outputs, forcing the decoder to adapt its reconstruction process to align with this translated space. As illustrated in Fig. 3, the FedOT$_{\text{tran}}$ method introduces severe color shifts after training the VAE encoder, which significantly degrades the quality of the generated images.

Mirror Transformation. For Gaussian distributions, applying a mirror operator $z' = -z$ to the latent variable z mirrors the probability distribution across the origin. Under this transformation, the new latent variable z' follows:

$$z' \sim \mathcal{N}(-\mu, \sigma^2), \tag{8}$$

which preserves the Gaussian variance structure while inverting the mean. We designate this approach as $\mathbf{FedOT_{mir}}$. Compared with FedOT$_{\text{tran}}$, the mirror transformation performs a symmetric reflection of the entire latent space. During Stage I, the VAE encoder learns to map inputs to this inverted latent representation. Subsequently, in Stage II, the decoder attempts to adapt its reconstruction to align with this mirrored space.

We observe that mirroring the latent space does not correspond to a simple semantic inversion (*e.g.*, color negation) in the pixel domain. Instead, it introduces severe distortions, particularly in color fidelity and fine-grained textures. As shown in Fig. 3, this process often leads to color inversion and blurring, significantly degrading the image quality after training the VAE encoder.

Negative Transformation. To preserve high-frequency details while enforcing a structural shift, we study pixel-domain inversions for latent transformation. Unlike direct latent operations, this strategy operates on the input data manifold, thereby implicitly reshaping the latent distribution while maintaining the Gaussian prior. Specifically, we introduce a deterministic pixel-wise inversion, denoted as $x^- = 1 - x$, and compel the VAE to adapt this mapping for normalized inputs. During Stage I, the encoder is trained to map the standard input x to a latent representation that corresponds to its negative counterpart x^-. Conversely, the decoder is tasked with reconstructing the original positive image x from this inverted latent code. This approach effectively implements a pixel negation within the latent space. We designate this method $\mathbf{FedOT_{neg}}$. As shown in Fig. 3, compared to the reflection in $\text{FedOT}_{\text{mir}}$, this pixel-guided strategy preserves more structural details and avoids the edge blurring artifacts.

Crucially, these latent transformations render the watermarked VAE necessary to the generative pipeline. Any attempt by a malicious client to replace the VAE with a clean counterpart will inevitably fail, as the U-Net has been trained to depend on the modified latent distribution. Without knowledge of the specific transformation parameters, such unauthorized replacements result in a mismatch between the latent space and the U-Net, leading to severe degradation of image synthesis quality. More implementation and training details are provided in Appendix C.

4 Experiments

4.1 Experimental Setup

Datasets. We leverage the COCO2017 [22] dataset to train the latent vector transformations of the VAE and embed the watermark, while utilizing LAION-10K [40] to simulate private data for federated fine-tuning of Stable Diffusion. Specifically, we employ a subset of COCO2017 comprising 10,000 diverse images (resized to 512×512) spanning 80 categories. For the federated scenario, we curate LAION-10K, a subset of LAION [38] containing 10,000 high-quality image-text pairs. Consistent with the protocol in IET [24], these images are resized to 256×256 for efficient local training.

Metrics. To comprehensively evaluate the performance of our proposed FedOT framework, we employ multiple widely used metrics. We use FID [14], SSIM [45], and PSNR to evaluate VAE reconstruction. FID and CLIP-Score [34] assess Stable Diffusion generation quality. Detection Rate and Bit Accuracy are used to reflect the reliability of watermark extraction and its effectiveness.

Implementation Details. Training diffusion models in a federated setting is challenging, as the U-Net parameters are continuously updated and latent-space-based watermarks are unsuited for this task. Therefore, we adopt a VAE-based Stable Signature [9] as our baseline. For training the LVT on the VAE, we set $\lambda_{KL} = 10^{-8}$ and the translation coefficient for $\text{FedOT}_{\text{tran}}$ to 11. During watermark training, $\lambda_i = 0.2$. For federated fine-tuning, we deploy Stable Diffusion

Table 1: Generation quality and comparison with Stable Signature* on 256×256 images and 48-bit watermarks. The left table reports results before the VAE replacement attack, while the right table shows performance after the attack. Changes in generated image quality before and after the attack are indicated as (green) for increases and (red) for decreases. Stable Signature* refers to applying Stable Signature watermarking within federated learning.

Method	Original				VAE Replacement Attack			
	FID↓	CLIP-Score↑	Detection↑	Bit Acc↑	FID↑	CLIP-Score↓	Detection↑	Bit Acc↑
Original SD [36]	22.99	0.316	–	–	–	–	–	–
Stable Signature* [9]	16.412	0.314	0.994	0.965	16.282 (-0.130)	0.312 (-0.002)	0.000	0.543
$\text{FedOT}_{\text{w/o LVT}}$	16.687	0.313	0.998	0.972	17.054 (+0.367)	0.311 (-0.002)	0.000	0.494
$\text{FedOT}_{\text{rand}}$	35.585	0.284	0.999	0.980	72.810 (+37.225)	0.260 (-0.024)	0.000	0.536
$\text{FedOT}_{\text{tran}}$	22.427	0.301	0.988	0.950	92.497 (+70.070)	0.268 (-0.033)	0.000	0.475
$\text{FedOT}_{\text{mir}}$	21.475	0.293	0.962	0.926	70.622 (+49.147)	0.229 (-0.064)	0.000	0.522
$\text{FedOT}_{\text{neg}}$	20.367	0.295	0.954	0.922	40.537 (+20.170)	0.272 (-0.023)	0.000	0.524

v2.1 [36] across $K = 5$ clients using the LAION-10K dataset, partitioned in an independent and identically distributed (i.i.d.) manner. Each client model undergoes local fine-tuning for 15 epochs, with 2,000 steps per epoch. The watermark length is set to $n = 48$ bits, partitioned into a prefix of $r = 16$ bits for global ownership verification and a suffix of 32 bits for precise client tracking. The detection threshold τ is set to 0.69, resulting in a FPR of 0.1%. We define attack failure as post-attack FID exceeding the original SD baseline (i.e., 22.99 FID as in Table 1), meaning all FL fine-tuning benefits are destroyed. With 4 RTX 4090, LVT training takes ∼24 h; training an FL client requires ∼7.5 h/GPU (15 epochs); training a watermarked VAE takes ∼5 min/GPU.

4.2 Main Results

Quantitative Analysis of Federated Fine-Tuning. Table 1 reports the results of fine-tuning Stable Diffusion under the FedOT framework. The Original SD baseline refers to the pre-trained Stable Diffusion V2 model evaluated on the LAION-10K validation set using FID and CLIP-Score. We adapt Stable Signature [9], originally designed for centralized training, to the federated setting as a comparison baseline, denoted as Stable Signature*. $\text{FedOT}_{\text{w/o LVT}}$ represents our method without applying the Latent Vector Transformation (LVT) strategy.

As shown in Table 1, the watermarks in both the baseline Stable Signature* and $\text{FedOT}_{\text{w/o LVT}}$ are not detected after the replacement attack, while the image quality remains almost unaffected. This indicates that the attack removes the watermark at little cost. In contrast, our method with the proposed LVT strategy experiences a noticeable degradation in image quality after the attack. This demonstrates LVT's effectiveness in preventing watermark removal attacks.

Performance of LVT under FedOT. We analyze different LVT strategies in Table 1. $\text{FedOT}_{\text{rand}}$ exhibits strong resistance to VAE replacement (+37.225) but suffers from severe degradation in generation quality (FID: 35.585). This supports our view that the VAE latent space is difficult to adapt to a randomly sampled Gaussian distribution. $\text{FedOT}_{\text{tran}}$ demonstrates strong resistance to VAE replacement (+70.070), but with lower image quality (FID: 22.427). This result

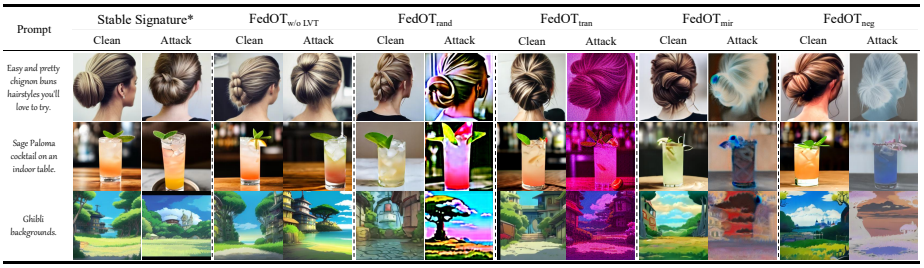

Fig. 4: Comparison between different FedOT methods and Stable Signature*. "Clean" represents the results without VAE replacement attacks, while "Attack" represents the results after VAE replacement attacks. This figure corresponds to Table 1.

reveals a critical trade-off: introducing a large translation coefficient induces a significant structural shift in the latent space, which effectively binds the components against replacement attacks but inevitably degrades generative fidelity. In comparison, FedOT$_{mir}$ exhibits similarly robust binding capability (+49.147) and offers improved generation quality (FID: 21.475). However, a critical observation is its severe impact on semantic consistency: after the replacement attack, the CLIP-Score drops significantly (-0.064). This indicates that the mirror transformation disrupts the semantic alignment between the text prompts and the generated images, rendering the stolen model practically useless for content creation. Finally, FedOT$_{neg}$ achieves the optimal balance, maintaining superior generation quality (FID: 20.367) while ensuring strong defense (+20.170). By leveraging pixel-domain inversion, this strategy preserves necessary high-frequency details while enforcing the latent shift. Unlike the translation and mirror transformations, it sustains a more consistent semantic structure, proving that carefully designed transformations can enhance watermark protection without sacrificing quality.

Visual Impact of Attacks. Fig. 4 visually compares the generative outputs under VAE replacement attacks. Consistent with Table 1, Stable Signature* and FedOT$_{\text{w/o LVT}}$ produce high-fidelity images post-attack, confirming their vulnerability to watermark removal without utility loss. In contrast, FedOT variants with LVT show visible degradation, demonstrating defense effectiveness. FedOT$_{\text{tran}}$ induces perceptible global color shifts, while FedOT$_{\text{neg}}$ manifests a clear luminance inversion. Notably, FedOT$_{\text{mir}}$ results in severe, unstructured distortions that disrupt semantic coherence, rendering the generated content visually unrecognizable and practically unusable for malicious actors.

Ownership Verification and Tracing. As shown in Table 2, Stable Signature* achieves ownership verification but lacks client traceability. In contrast, our FedOT framework supports both, due to its watermark design. Among variants, FedOT$_{\text{w/o LVT}}$ performs best in both metrics. However, this variant and Stable Signature* are vulnerable to replacement attacks, as shown in Table 1. With the introduction of LVT, the performance of FedOT$_{\text{tran}}$, FedOT$_{\text{mir}}$, and FedOT$_{\text{neg}}$ slightly decreases but remains robust. All variants maintain detection

Table 2: Comparison of Stable Signature* and the FedOT in the federated learning framework, focusing on ownership verification and tracing.

Method	Ownership		Tracing	
	Detection↑	Bit Acc↑	Detection↑	Bit Acc↑
Stable Signature*	0.994	0.965	–	–
$\text{FedOT}_{\text{w/o LVT}}$	0.996	0.981	0.993	0.967
$\text{FedOT}_{\text{tran}}$	0.983	0.962	0.975	0.945
$\text{FedOT}_{\text{mir}}$	0.972	0.953	0.935	0.913
$\text{FedOT}_{\text{neg}}$	0.960	0.947	0.932	0.910

Table 3: Performance of $\text{FedOT}_{\text{tran}}$ under varying translation coefficients c.

$\text{FedOT}_{\text{tran}}$	Original				VAE Replacement Attack			
	FID↓	CLIP-Score↑	Detection↑	Bit Acc↑	FID↑	CLIP-Score↓	Detection↑	Bit Acc↑
$c = 2$	22.656	0.303	0.959	0.940	21.232 (-1.424)	0.305 (+0.002)	0.000	0.507
$c = 5$	19.684	0.306	0.940	0.935	20.749 (+1.065)	0.305 (-0.001)	0.000	0.513
$c = 8$	21.461	0.305	0.959	0.939	23.752 (+2.291)	0.299 (-0.006)	0.000	0.512
$c = 11$	22.427	0.301	0.974	0.950	92.497 (+70.070)	0.268 (-0.033)	0.000	0.475

rates above 0.932 and bit accuracy over 0.91, demonstrating the effectiveness and necessity of our method.

4.3 Ablation Studies

Translation Coefficient c. We investigate the sensitivity of $\text{FedOT}_{\text{tran}}$ to the translation coefficient c. As expected, our results reveal a critical trade-off between generative fidelity and resistance to replacement attacks. Setting c to an excessively small or large value degrades the models' optimal performance. Interestingly, under a weak perturbation ($c = 2$), replacing the watermarked VAE with a clean counterpart actually results in improvement in image quality. This suggests that too small latent shifts fail to enforce component binding. Conversely, as c increases, the binding becomes increasingly stringent. When $c = 11$, the image quality degradation induced by a replacement attack surpasses even that of the $\text{FedOT}_{\text{mir}}$. These findings empirically demonstrate that a larger translation coefficient significantly amplifies the models' ability against attacks, but at the inevitable cost of generation quality.

Impact of Weighting Coefficient λ_i. We further perform an ablation study on the weighting coefficient λ_i. As shown in Table 4, varying λ_i introduces a direct tension between visual quality and watermark reliability. A larger λ_i prioritizes the reconstruction loss, compelling the generated images to closely resemble the original inputs. However, this heavily penalizes the watermark embedding loss, resulting in significantly lower bit accuracies during message extraction. Conversely, decreasing λ_i strengthens the watermark, ensuring bit accuracy.

Different Lengths of the Chunked Watermark r. In the FedOT framework, the watermark of length $n = 48$ is split into two segments: the first r bits are designated for ownership verification, while the remaining $n - r$ bits are assigned to client tracing. We explore how different values of r affect both tasks by conducting ablation experiments, with results reported in Table 5. When $r = 8$, fewer bits are available for ownership verification, leading to weaker performance (Detection: 0.944, Bit Acc: 0.928). When $r = 24$, the tracing performance noticeably drops (Detection: 0.928, Bit Acc: 0.899) due to fewer bits being available

Table 4: Trade-off between PSNR and Bit Accuracy under different λ_i

λ_i **for Fine-tuning**	0.8	0.4	0.2	0.1	0.05
PSNR ↑	34.793	34.223	33.611	32.953	29.842
Bit Acc ↑	0.906	0.927	0.935	0.960	0.965

Table 5: impact of different r on ownership verification and tracing.

r**-length**	**Ownership**		**Tracing**	
	Detection ↑	Bit Acc ↑	Detection ↑	Bit Acc ↑
$r = 8$	0.944	0.928	0.957	0.921
$r = 16$	0.960	0.947	0.932	0.910
$r = 24$	0.958	0.946	0.928	0.899

for client encoding. The setting $r = 16$ achieves the best balance, with strong performance in both ownership and tracing. This configuration is therefore used as the default in all other experiments.

4.4 Impact of Federated Conditions

FedOT with Different Numbers of Clients. We evaluate the performance of FedOT with 5, 10, and 20 clients, as shown in Table 6. Overall, the generation quality remains relatively stable as the number of clients increases. Meanwhile, watermark robustness stays largely unaffected across all settings, further demonstrating the effectiveness of our LVT-based watermarking approach. In all cases, detection rates exceed 0.941, indicating that even in federated learning environments with diverse and heterogeneous client datasets, the number of clients has a limited impact on the ability to reliably detect and trace watermarks.

FedOT under Non-i.i.d. Distributions. To assess robustness under realistic conditions, we simulate non-i.i.d. settings using Dirichlet distributions with varying α, as shown in Table 7. With 5 clients, all FedOT variants maintain stable generation quality and watermark robustness across different α values. In all cases, the detection remains above 0.957, indicating that data heterogeneity has little impact on watermark robustness.

4.5 Purification Attack

A purification attack targets watermarks or hidden information embedded in model parameters. In this scenario, the attacker fine-tunes or optimizes the model using a clean dataset without any watermark, aiming to "purify" the model by removing the watermark while preserving the model's original performance as much as possible. In other words, the attacker retrains part of the model with real, unwatermarked data to erase or weaken the previously embedded watermark without significantly degrading the model's output quality.

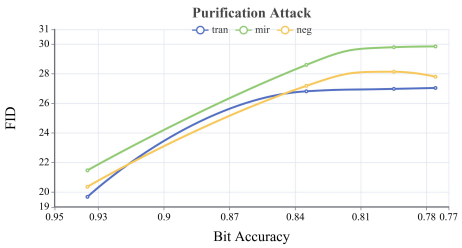

Fig. 5: FID variation during the purification process for 3 different FedOT methods.

Table 6: FID, CLIP-Score, Bit Accuracy, and Detection across different numbers of clients.

Method	Metric	Number of Clients		
		5	10	20
FedOT$_{\text{tran}}$ ($c = 5$)	FID ↓	19.684	20.119	19.449
	CLIP-Score ↑	0.306	0.306	0.304
	Bit acc ↑	0.935	0.941	0.940
	Detection ↑	0.968	0.983	0.978
FedOT$_{\text{mir}}$	FID ↓	21.475	22.540	22.22
	CLIP-Score ↑	0.293	0.293	0.291
	Bit acc ↑	0.926	0.928	0.926
	Detection ↑	0.962	0.970	0.967
FedOT$_{\text{neg}}$	FID ↓	20.367	20.541	20.23
	CLIP-Score ↑	0.295	0.294	0.293
	Bit acc ↑	0.922	0.914	0.910
	Detection ↑	0.953	0.947	0.941

Table 7: FID, CLIP-Score, Bit Accuracy, and Detection under different Dirichlet distributions.

Method	Metric	Data Distribution		
		i.i.d	$\alpha = 0.7$	$\alpha = 0.5$
FedOT$_{\text{tran}}$ ($c = 5$)	FID ↓	19.684	20.33	20.177
	CLIP-Score ↑	0.306	0.306	0.306
	Bit acc ↑	0.935	0.942	0.941
	Detection ↑	0.968	0.983	0.980
FedOT$_{\text{mir}}$	FID ↓	21.475	22.294	22.437
	CLIP-Score ↑	0.293	0.290	0.289
	Bit acc ↑	0.926	0.935	0.935
	Detection ↑	0.962	0.974	0.976
FedOT$_{\text{neg}}$	FID ↓	20.367	21.491	21.380
	CLIP-Score ↑	0.295	0.293	0.293
	Bit acc ↑	0.922	0.924	0.921
	Detection ↑	0.953	0.960	0.957

We conduct purification attacks by fine-tuning the watermarked parameter components using a clean dataset [22]. Fig. 5 illustrates the change in FID during the watermark purification process. After 300 epochs of purification, the FID of all three LVT methods increased from below 22 to above 26, with FedOT$_{\text{mir}}$ showing the largest increase. This demonstrates that removing the watermark inevitably impacts the quality of generated images, validating the robustness of our proposed methods. Empirically, it is very challenging to eliminate watermarks without sacrificing image quality. More robustness evaluations are provided in Appendix D.

5 Conclusion

This paper introduces **FedOT**, the first ownership verification and tracing framework for Latent Diffusion Models (LDMs) in Federated Learning. To identify the responsible client for a leak, we propose a chunked watermark for ownership verification and traceability. The chunked watermark addresses a critical gap in protecting and ensuring traceability for generative models within FL. Unlike existing VAE-based watermarking methods, which are vulnerable to removal attacks, FedOT leverages **Latent Vector Transformation (LVT)** to modify the latent space of the VAE, effectively binding the VAE and U-Net. Ensuring that any attempt to replace the VAE for watermark removal leads to severe degradation in image quality. We propose three LVT strategies, including translation, mirror, and negative transformation. Although the replacement attack can remove the watermark, it comes at the cost of a severe loss in image quality. This trade-off renders the attack impractical and demonstrates the robustness of our method.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62436007, No. 62572147)

References

1. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., et al.: ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
3. Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., et al.: The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. arXiv preprint arXiv:1802.07228 (2018)
4. Cao, Y., Li, S., Liu, Y., Yan, Z., Dai, Y., Yu, P., Sun, L.: A survey of ai-generated content (aigc). ACM Computing Surveys (2025)
5. Ci, H., Song, Y., Yang, P., Xie, J., Shou, M.Z.: Wmadapter: Adding watermark control to latent diffusion models. In: ICML (2025)
6. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital watermarking and steganography. Morgan Kaufmann (2007)
7. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
8. Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Möllering, H., Nguyen, T.D., Rieger, P., Sadeghi, A.R., Schneider, T., Yalame, H., et al.: Safelearn: Secure aggregation for private federated learning. In: IEEE Security and Privacy Workshops (2021)
9. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. In: ICCV (2023)
10. Gan, Y., Miao, J., Wang, Y., Yang, Y.: Silence is golden: Leveraging adversarial examples to nullify audio control in ldm-based talking-head generation. In: CVPR (2025)
11. Gan, Y., Miao, J., Yang, Y.: Datastealing: Steal data from diffusion models in federated learning with multiple trojans. In: NeurIPS (2024)
12. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR (2022)
13. Han, L., Li, Y., Zhang, H., Milanfar, P., Metaxas, D., Yang, F.: Svdiff: Compact parameter space for diffusion fine-tuning. In: ICCV (2023)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
16. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: NeurIPS (2022)
17. Hu, R., Zhang, J., Li, Y., Li, J., Guo, Q., Qiu, H., Zhang, T.: Videoshield: Regulating diffusion-based video generation models via watermarking. In: ICLR (2025)

18. Jia, H., Zhao, N., Xu, Y., Zhu, L., Yang, Y.: Gas: Geometry-appearance synergy for consistent video customization. In: MMM (2026)
19. Lei, L., Gai, K., Yu, J., Zhu, L.: Diffusetrace: A transparent and flexible watermarking scheme for latent diffusion model. arXiv preprint arXiv:2405.02696 (2024)
20. Li, B., Fan, L., Gu, H., Li, J., Yang, Q.: Fedipr: Ownership verification for federated deep neural network models. IEEE TPAMI (2022)
21. Li, D., Xie, W., Wang, Z., Lu, Y., Li, Y., Fang, L.: Feddiff: Diffusion model driven federated learning for multi-modal and multi-clients. IEEE TCSVT (2024)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
23. Liu, G., Cao, S., Qian, Z., Zhang, X., Li, S., Peng, W.: Watermarking one for all: A robust watermarking scheme against partial image theft. In: CVPR (2025)
24. Liu, X., Guan, X., Wu, Y., Miao, J.: Iterative ensemble training with anti-gradient control for mitigating memorization in diffusion models. In: ECCV (2024)
25. Liu, X., Shao, S., Yang, Y., Wu, K., Yang, W., Fang, H.: Secure federated learning model verification: A client-side backdoor triggered watermarking scheme. In: IEEE International Conference on Systems, Man, and Cybernetics (2021)
26. Ma, Z., Jia, G., Qi, B., Zhou, B.: Safe-sd: Safe and traceable stable diffusion with text prompt trigger for invisible generative watermarking. In: ACM MM (2024)
27. McMahan, B., Moore, E., Ramage, D., Hampson, S.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS (2017)
28. Min, R., Li, S., Chen, H., Cheng, M.: A watermark-conditioned diffusion model for ip protection. In: ECCV (2024)
29. Morafah, M., Reisser, M., Lin, B., Louizos, C.: Stable diffusion-based data augmentation for federated learning with non-iid data. arXiv preprint arXiv:2405.07925 (2024)
30. Morita, R., Frolov, S., Moser, B.B., Shirakawa, T., Watanabe, K., Dengel, A., Zhou, J.: Tkg-dm: Training-free chroma key content generation diffusion model. In: CVPR (2025)
31. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2022)
32. Nightingale, S.J., Farid, H.: Ai-synthesized faces are indistinguishable from real faces and more trustworthy. PNAS (2022)
33. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. In: ICLR (2024)
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
35. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
37. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: CVPR (2023)
38. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)

39. Shao, S., Yang, W., Gu, H., Qin, Z., Fan, L., Yang, Q.: Fedtracker: Furnishing ownership verification and traceability for federated learning model. IEEE TDSC (2024)
40. Somepalli, G., Singla, V., Goldblum, M., Geiping, J., Goldstein, T.: Understanding and mitigating copying in diffusion models. In: NeurIPS (2023)
41. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
42. Stanley Jothiraj, F.V., Mashhadi, A.: Phoenix: A federated generative diffusion model. In: WWW (2024)
43. Tekgul, B.G., Xia, Y., Marchal, S., Asokan, N.: Waffle: Watermarking in federated learning. In: International Symposium on Reliable Distributed Systems (2021)
44. Tian, Z., Quan, R., Ma, F., Zhan, K., Yang, Y.: Brainguard: Privacy-preserving multisubject image reconstructions from brain activities. In: AAAI (2025)
45. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP (2004)
46. Wang, Z., Guo, J., Zhu, J., Li, Y., Huang, H., Chen, M., Tu, Z.: Sleepermark: Towards robust watermark against fine-tuning text-to-image diffusion models. In: CVPR (2025)
47. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-rings watermarks: Invisible fingerprints for diffusion images. In: NeurIPS (2023)
48. Xia, C., Ma, F., Quan, R., Xu, Y., Zhan, K., Yang, Y.: Echoes of ownership: Adversarial-guided dual injection for copyright protection in mllms. In: CVPR (2026)
49. Xu, Y., Zhu, L., Yang, Y.: Gg-editor: Locally editing 3d avatars with multimodal large language model guidance. In: ACM MM (2024)
50. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM TIST (2019)
51. Yang, W., Shao, S., Yang, Y., Liu, X., Liu, X., Xia, Z., Schaefer, G., Fang, H.: Watermarking in secure federated learning: A verification framework based on client-side backdooring. ACM TIST (2023)
52. Yang, Y., Zhuang, Y., Pan, Y.: Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. Frontiers of Information Technology & Electronic Engineering (2021)
53. Yang, Z., Zeng, K., Chen, K., Fang, H., Zhang, W., Yu, N.: Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In: CVPR (2024)
54. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: CVPR (2021)
55. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
56. Zhou, D., Li, M., Yang, Z., Lu, Y., Xu, Y., Wang, Z., Huang, Z., Yang, Y.: BideDPO: Conditional image generation with simultaneous text and condition alignment. In: ICLR (2026)
57. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: ECCV (2018)