Towards Unified Scalable Watermarking of Generative Models for User-Wise Attribution

Ziqi Yang [*1] **Jun Yu** [*1] **Ruite Xu** [1] **Kui Ren** [1]

Abstract

The proliferation and potential misuse of Generative AI (GenAI), coupled with the inherent reluctance of companies to deploy watermarking mechanisms, underscores the urgent need for a **unified scalable paradigm** for attributing policy-violating images to responsible users. Unfortunately, existing model watermarking methods either lack generalizability due to their dependence on specific architectures, or lack scalability for the growing number of large models and users. To approach practical deployment, we propose **the first gray-box and practically scalable model watermarking framework** for user-wise attribution, namely ***Unimark***. Specifically, we introduce a trustworthy third party, the *regulator*, to unify all watermarks with a pre-trained universal watermark extractor. The regulator trains a mapping model for each generative model, which learns to adversarially perturb the model weights via only a low-dimensional proxy to embed a user-specific watermark. Extensive experiments demonstrate that our framework generalizes well across 4 state-of-the-art generative model architectures enabling unified attribution across them, and scales well with increasing number of models, users and images. Moveover, our approach aligns with baselines on fidelity and robustness. Our results not only suggest the effectiveness and practicality of the proposed algorithm, but also **shed light on the possibility of a unified regulation paradigm for GenAI**.

1. Introduction

Generative AI (GenAI) (Sauer et al., 2022; Rombach et al., 2022; Song et al., 2020; Ramesh et al., 2022; Nichol et al., 2021) have flourished in recent years, attracting significant attention for their exceptional performance. Users can download the model weights or access the API of these prevailing generative models, and then generate images under an **usage policy**, e.g. OpenAI Usage Policies (OpenAI, b) and Midjourney Terms of Service (Midjourney). This comes up with a threat that one may violate the policy and use the generative models to generate disapproved content, which raises the need to detect (Marra et al., 2019; Rossler et al., 2019; Li et al., 2020; Qian et al., 2020; Wu et al., 2020; Frank et al., 2020; Hu et al., 2021b; Jeong et al., 2022) and attribute (Yu et al., 2020; Fernandez et al., 2023; Kim et al., 2023; Lukas & Kerschbaum, 2023) AI-generated content to its responsible user. This user-wise attribution is challenging but crucial for legal adjudication and deterrence of model misuse. Watermarking, as a proactive attribution approach, has gained substantial international attention. Companies including Google (Google DeepMind), Meta (Fernandez et al., 2023) and OpenAI (OpenAI, a) have been actively investigating and developing it, while some governments regulations mandate its use (The White House).

Despite international attention and active research, we note that GenAI companies are **inherently reluctant** to deploy user-wise watermarking mechanisms, as it provides no direct revenue for companies and, on the contrary, may significantly reduce their user base: a survey (Seetharaman & Barnum, 2024) revealed that nearly 30% of ChatGPT users indicated they would reduce their usage of the service if ChatGPT deployed watermarks while a rival did not. Moreover, the **limited generalizability and inadequate scalability** of existing watermarking methods further hinder their deployment.

To be specific, we highlight the emerging trend of **model watermarking** (Yu et al., 2020) for user-wise attribution, where user-specific watermarks are embedded in the model weights so that all subsequently generated content contains the watermark. It achieves sota scalability across images by merging the watermarking process into the generation process and enhances robustness by protects the watermark from trivial removal attempts even in the open-source scenario. However, existing studies in this paradigm **lack generalizability to emerging diverse model architectures** due to their dependence on a specific model architecture by design (e.g., GANs (Yu et al., 2020; Lukas & Kerschbaum, 2023) and Latent Diffusion Models (Fernandez et al., 2023;

[*]Equal contribution [1]Zhejiang University. Correspondence to: Ziqi Yang <yangziqi@zju.edu.cn>.

Preprint, 2025. Copyright 2025 by the author(s).

Kim et al., 2023)). On the other hand, **the real-world efficiency and scalability of existing work are neither sufficient**. Most existing studies (Kim et al., 2023; Lukas & Kerschbaum, 2023; Yu et al., 2020) necessitates training a separate watermark extractor for each model, which is hard to generalize and costly to scale. Other studies (Fernandez et al., 2023) with a single extractor necessitates a separate fine-tuning process for each user, which is also hard to scale due to the linear increase of computational costs with the number of users.

To approach practical deployment, we envision a **unified watermarking paradigm**, where a *trustworthy third party* (e.g., regulatory authority) can perform standardized large-scale watermarking with only *grey-box access* to the model.

We propose ***Unimark***, the *first model-agnostic and practically scalable model watermarking framework* for watermarking vision generative models in a unified manner. We introduce a third party, termed the *regulator*, to decouple watermarking processes from (unmotivated) model inventors, and *unify* them across different inventors and models with a pretrained universal watermark extractor. We decompose the model watermarking process into two stages: *mapping* and *decoding*. The mapping stage, conducted by the regulator, employs a model-specific *mapping model* (MLP in this work) to generate a compact set of encoded parameter perturbation for each user. Subsequently, in the decoding stage, the inventor, with access to the original model weights, decodes the encoded perturbation back into parameter space to get watermarked model. The encoded parameter perturbation acts as a *proxy* or *universal interface* for the regulator to perform watermarking.

We identify novel real-world challenges in our framework and propose solutions to address each of them. Our framework demands higher *computational* and *memory* efficiency and introduces novel aspects of efficiency, particularly in inter-party *communication cost*. Moreover, it demands *minimal information disclosure* between the regulator and inventor to protect both parties' interests, particularly the inventor's model intellectual property. To reduce inter-party communication cost and potential training runs needed for tuning hyperparameters, we design a low-dimensional proxy and a bounded adversarial training algorithm. To minimize information disclosure (for protecting model IP in particular), we design the training algorithm to be gray-box and the proxy to be structureless and shuffled.

We conduct extensive experiments to validate the superior generalizability and scalability of our approach and its maintainance of state-of-the-art fidelity and robustness. Specifically, our framework generalizes well across 4 different state-of-the-art generative model architectures and enables unified attribution across different inventors/models via the universal watermark extractor, while requiring only one forward pass of mapping and decoding to infer a watermarked model. This represents scalability and generalizability across generated images, users, model architectures, and inventors. Moveover, our approach aligns with state-of-the-arts baselines by achieving 99% attribution accuracy and comparable robustness with overall FPR set to 0.01% and minimal FID loss.

Technical Contributions. In this work, we watermark generative models in a unified manner for user-wise attribution. Our main contributions are in four thrusts:

- We point out that despite strong advocacy from regulatory bodies, GenAI companies are **inherently reluctant** to deploy user-wise attribution mechanisms. Moreover, the **low generalizability and insufficient scalability** of existing watermarking schemes further hinder their deployment.

- We envision a unified watermarking paradigm to mitigate this problem and propose **Unimark**, the *first model-agnostic and practically scalable model watermarking framework*, which unburdens model inventors from watermark-related processes and introduces the *regulator* to assume the responsibility and unify the processes.

- We identify novel real-world challenges in our framework and propose solutions to address each of them. To reduce inter-party communication cost and redundant training runs needed for tuning hyperparameters, we design a low-dimensional encoded parameter perturbation as proxy and a bounded adversarial training algorithm. To minimize information disclosure (for protecting model IP in particular), we design the training algorithm to be gray-box and the proxy to be structureless and shuffled.

- We conduct extensive experiments to validate the superior generalizability and scalability of our approach and its maintainance of state-of-the-art fidelity and robustness. Our method generalizes well across 4 different state-of-the-art generative model architectures and achieves 99% attribution accuracy for each model and the mixed test when under cropping or no attack. Our method scales well to increasing number of users and models, with the mapping model training costs only 1 H100 GPU hour for SDXL-Turbo, which is 100x cheaper than existing methods and enables inference of a watermarked model with only one forward pass. Our watermark has a minimal effect on FID and is comparable to baselines.

2. Background

Prior paradigm of user-wise model watermarking involves two parties: the model **inventor**, who develops and commercializes advanced generative models, and the **user**, who generates images under an *usage policy* with watermarked models. The latter is considered *untrusted* and may attempt model misuse or watermark removal. Therefore, the watermarking process is assumed to be entirely conducted by the inventor, i.e., *inventor-oriented* (Fernandez et al., 2023; Lukas & Kerschbaum, 2023; Kim et al., 2023).

Given a pre-trained generative model $\mathcal{G}_\theta : \mathbb{X} \mapsto \mathbb{Y}$ parameterized by θ, where $\mathbb{X}$ represents the input space and $\mathbb{Y}$ represents the image space, and a user-specific watermarking message $m_i \in \{0,1\}^{d_m}$ of bit length d_m for user u_i, the inventor incorporates m_i into θ through a parameter watermarking process Θ (e.g., fine-tuning (Fernandez et al., 2023) or modulation (Lukas & Kerschbaum, 2023; Kim et al., 2023)):

$$\theta_i = \Theta(\theta, m_i). \tag{1}$$

Subsequently, given an input $x \in \mathbb{X}$, the user-specific model $\mathcal{G}_{\theta_i}$ with the watermarked parameters θ_i can generate watermarked images $y_i = \mathcal{G}_{\theta_i}(x)$ that carry the message m_i. A watermark extractor $\mathcal{W} : \mathbb{Y} \mapsto \{0,1\}^{d_m}$ extracts the message $m_i' = \mathcal{W}(y_i)$ from y_i, which is expected to be the same as m_i, meaning their distance $L_w(m_i, m_i')$ (e.g., ℓ_2 distance) is minimized. Moreover, y_i is expected to have the similar image quality with its vanilla version $y = \mathcal{G}_\theta(x)$, meaning their perceptual distance $L_p(y, y_i)$ (e.g., LPIPS (Zhang et al., 2018)) is constrained by a perceptual deterioration limit $\epsilon \in \mathbb{R}^+$ **(fidelity)**. The overall watermarking objective writes:

$$\min_{\Theta} \mathbb{E}_{x,i}[L_w(m_i, m_i')] \quad s.t. \quad \mathbb{E}_{x,i}[L_p(y, y_i)] \leq \epsilon. \tag{2}$$

Moreover, the watermark is expected to be robust in the sense that even if y_i is manipulated by image editing transformations such as cropping, resizing and jpeg compression, or even the model $\mathcal{G}_{\theta_i}$ is altered, $\mathcal{W}$ is still able to accurately extract the watermark message m_i **(robustness)**.

Yu et al. (Yu et al., 2020) train GANs from scratch with modulation as watermarking. PTW (Lukas & Kerschbaum, 2023) applies modulation on pretrained GANs to improve efficiency. WOUAF (Kim et al., 2023) migrate this idea to LDMs. However, these methods require training a separate extractor for each model. Stable Signature (Fernandez et al., 2023) proposes to watermark LDMs (and subsequently images generated by it) by fine-tuning the model against a pre-trained watermark decoder with a specific watermarking message for each user. However, its computational demand increase linearly with the number of users since it necessitates a fixed-time fine-tuning process for each individual. In contrast, our method achieves fixed training costs regardless of user growth and generalizes well across different model architectures.

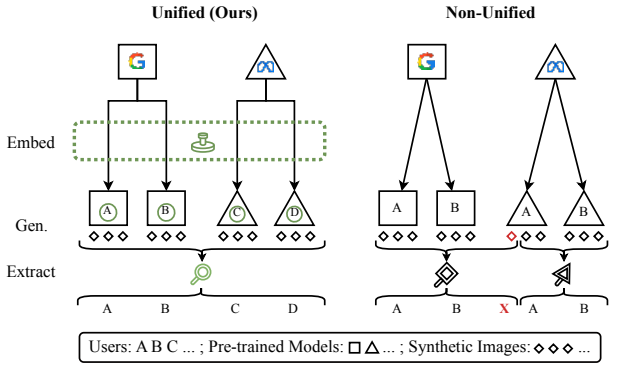

Figure 1. **Unified User-Wise Model Watermarking.** The green color indicates the unified parts: a unified embedding process, universal watermark carriers (as a result), and a universal extractor. We compare our setting with existing non-unified settings, where each model (Yu et al., 2020; Lukas & Kerschbaum, 2023; Kim et al., 2023) or each inventor (Fernandez et al., 2023) has its own watermarking scheme, which leads to computational overhead and potential cross-attribution error (e.g., red X in the figure).

3. Our Approach

3.1. Overview

We present an overview of our framework in Figure 2. Based on the prior paradigm, we envision a unified user-wise model watermarking paradigm to facilitates regulation and development of generative models.

Unified Paradigm. We introduce a *trusted third party*, the **regulator**, to *decouple* the watermarking process from inventors and *unify* it across different inventors/models, e.g. an official agency, who offers model watermarking service to inventors, empowering any generative model from a inventor to generate images born with user-specific watermarks, and regulates the subsequent generated content (via extracting and verifying watermarks in policy-violating content).

To decouple the watermarking process from inventors, we decompose the watermarking process in Equation (1) into two parts:

$$\theta_i = (\mathcal{F}_\theta^{-1} \circ \mathcal{M})(m_i), \tag{3}$$

where a mapping function $\mathcal{M}$ (for each $\mathcal{G}_\theta$, at regulator side) maps the message m_i to a much smaller-sized set of user-specific parameters $\psi_i = \mathcal{M}(m_i)$ with $|\psi_i| \ll |\theta_i|$, that we coin *proxy*, and a proxy decoder $\mathcal{F}_\theta^{-1}$ (with access to θ, at inventor side) decodes ψ_i to $\theta_i = \mathcal{F}_\theta^{-1}(\psi_i)$. We assign a *universal extractor* $\mathcal{W}$, which is applicable across all inventors/models, to the regulator as a secret. In practice, we initially design the encoding function $\mathcal{F}_\theta$ that maps θ_i to ψ_i, subsequently derive its inverse $\mathcal{F}_\theta^{-1}$, and finally train

the corresponding $\mathcal{M}$.

This framework facilitates unified regulation as shown in Figure 1, in which the regulator sees all models as *gray boxes* and has no direct access to the original model. This approach ensures a higher level of abstraction and generalizability across various models and inventors.

Pretrained Robust Extractor as Universal Extractor. Prior works achieve robustness through three ways: (i) training-time perturbations (Zhu et al., 2018; Lukas & Kerschbaum, 2023; Kim et al., 2023), (ii) high-level semantic features (Wen et al., 2023) and (iii) optimizing against pretrained robust extractors (Fernandez et al., 2023). Following the last trend, for consistency, we adopt the extractor from StableSignature as the universal extractor by default (a whiten 48-bit HiDDeN (Zhu et al., 2018) CNN decoder, more detials in Appendix B.3), for its unbiasedness and robustness. Our mapping model is adversarially trained against the extractor, to maintain the robustness (see Section 4.5).

Mapping Model. For simplicity, we implement the mapping function $\mathcal{M}$ as a multi-layer perceptron (MLP) $\mathcal{M}_\Phi$ parameterized by Φ and refer to it as the *mapping model*. One $\mathcal{M}_\Phi$ corresponds to one $\mathcal{G}_\theta$. Since it maps m_i to ψ_i, we initialize it with an input dimension of d_m (message bit length), an output dimension of d_ψ (proxy parameter size), and one hidden layer of dimension 1024 by default. The mapping model, combined with the proxy decoding function $\mathcal{F}_\theta^{-1}$, can be seen as a hypernetwork (Ha et al., 2016) that generates the weights of a watermarked generative model based on the user-specfic message.

Following objective (2), we train $\mathcal{M}_\Phi$ to ensure that the watermarked model $\mathcal{G}_{\theta_i}$ generates watermarked images y_i whose visual quality remains high (via constraining their L_p distance with their vanilla versions y), and from whom watermarks can be extracted accurately (via minimizing $L_w(m_i, m_i')$). Following Fernandez et al., we use Watson-VGG perceptual loss (Czolbe et al., 2020) (an improved version of LPIPS (Zhang et al., 2018)) as L_p. It quantifies a perceptual similarity between vanilla generated images and watermarked images using deep feature extractors. And we use the Binary Cross Entropy (BCE) as L_w. Finally, the task of finding Θ in Equation (2) becomes optimizing over Φ:

$$\min_{\Phi} \mathbb{E}_{x,i}[L_w(m_i, m_i')] \quad s.t. \quad \mathbb{E}_{x,i}[L_p(y, y_i)] \leq \epsilon. \quad (4)$$

We train it in a gray-box and quality-bounded manner through the collaborative efforts of the regulator and inventor (Section 3.3). To improve scalability given the large size of θ, we make it parameter-efficient (and subsequently computation-efficient, memory-efficient, communication-efficient, etc), by encoding θ_i into the low-dimensional proxy ψ_i using a carefully chosen $\mathcal{F}_\theta$ (Section 3.2).

3.2. Perturbation Encoding

Drawing inspiration from parameter-efficient fine-tuning (PEFT) literature (Hu et al., 2021a; Si et al.), we encode the user-specific model perturbation into a low-dimensional vector - the user-specific encoded perturbation $\psi_i = \mathcal{F}_\theta(\theta_i)$, where $\mathcal{F}_\theta$ is the encoding function processed by inventor. ψ_i serves as a proxy or universal interface between the regulator and inventor, which also reduces the communication cost and information disclosure. Conceptually all PEFT or compression methods can be applied here. In our work, we design the encoding function $\mathcal{F}_\theta$ as a pipeline of five operations: $\mathcal{F}_\theta = (\text{PRUNE}; \text{POOL}; \text{DECOMPOSE}; \text{FLATTEN}; \text{PERMUTE})$. For clarity, we consider only convolutional layers in this section.

For each convolutional layer with relative perturbation $\Delta W = \frac{W - W_0}{W_0} \in \mathbb{R}^{c_{out} \times c_{in} \times k \times k}$ where W and W_0 are the watermarked and original weights respectively, we downsample it by average pooling each kernel: $\Delta W_p = \mathbb{E}_{i,j}[\Delta W[:, :, i, j]] \in \mathbb{R}^{c_{out} \times c_{in}}$. We then approximate this tensor with a low-rank decomposition: $BA \approx \Delta W_p$ where $B \in \mathbb{R}^{c_{out} \times r}$, $A \in \mathbb{R}^{r \times c_{in}}$, and rank $r \ll min(c_{out}, c_{in})$. We flatten and concatenate all B and A into a single vector v_i. Finally, we shuffle this vector to generate the final $\psi_i = Pv_i$, where P is a permutation matrix randomly sampled in advance. The previous steps is mainly for dimension reduction and the final step is for reducing information disclosure (see Section 4.4). Note that the permutation has no impact on performance, as MLPs ($\mathcal{M}_\Phi$) exhibit permutation invariance.

3.3. Gray-box Constrained Training

The regulator offers model watermarking service to model inventors, who are typically reluctant to 1) disclose their model parameters or 2) compromise the quality of generated images. To this end, we implement gray-box constrained training between the regulator and a inventor with *limited information exchange* and *bounded quality*. We formalize the process in Algorithm 1 and show the corresponding data flow in Figure 3.

We achieve hard-bounded quality by constraining the perceptual loss $\mathcal{L}_p$ within a threshold ϵ in step 7-16 before each optimization step of $\mathcal{L}_w$. This reduces redundant training runs for tuning hyperparameters, e.g. the balancing hyperparameter in prior art (Kim et al., 2023; Fernandez et al., 2023). Note that certain steps in the algorithm can be reordered and executed in parallel to further enhance computational efficiency. We omit further implementation for simplicity.

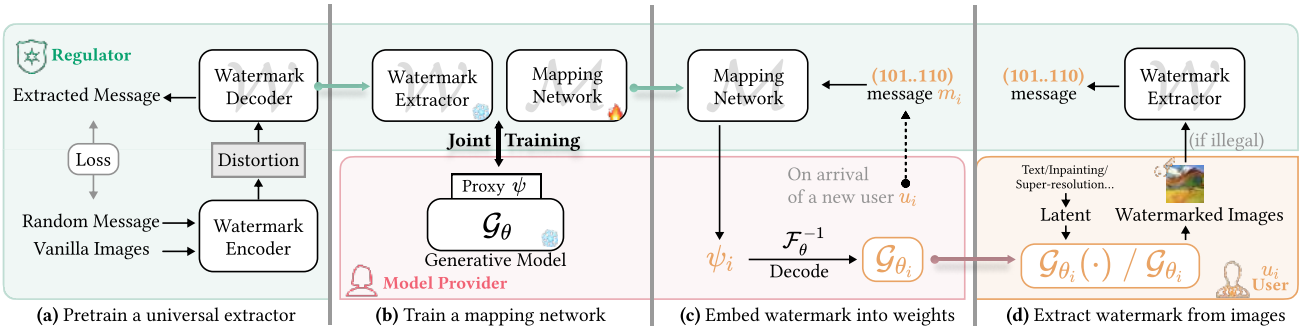

Figure 2. An overview of Unimark. (a) The regulator pre-trains a model-agnostic universal extractor $\mathcal{W}$, which extracts binary messages robustly. (b) The regulator then co-trains a mapping model $\mathcal{M}_\Phi$ for each model $\mathcal{G}_\theta$ from a model inventor, which learns to adversarially perturb the model weights for watermarking. At deployment stage, they (c) embed a user-specific message m_i into model weights for each user u_i before offering API/weights, and (d) attribute policy-violating content to responsible users by extracting and matching the embedded messages.

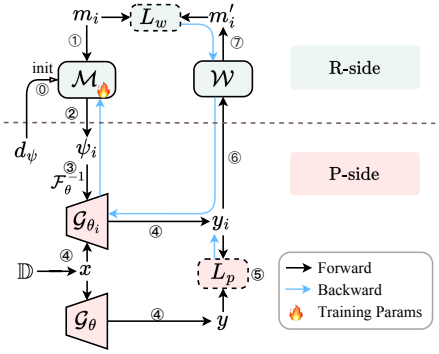

Figure 3. Data Flow of Gray-box Constrained Training.

4. Experimental Evaluation

4.1. Setup

Pre-trained Models. We conducted evaluations on four state-of-the-art generative models representing four different model architectures: Latent Diffusion Model (LDM) represented by SDXL-Turbo (Sauer et al., 2023), Latent Consistency Model (LCM) represented by LCM-XL (Luo et al., 2023), Diffusion Transformer (DiT) represented by DiT-XL-2-512 (Peebles & Xie, 2022), and Generative Adversarial Network (GAN) represented by StyleGAN-XL (Sauer et al., 2022). We present the details of these models in Appendix B.3. For text-to-image models (LDM/LCM), we utilize MS-COCO (Lin et al., 2014) as the dataset, which contains 83k images in the training set and 41k images in the validation set, each annotated with 5 captions. We train the mapping model on the training set, and then evaluate with the validation set. During evaluation, we set output image resolutions to each generator's recommended specifications (mainly 512×512, see Appendix B.3) and generate images using random MS-COCO captions. For label-to-image models (DiT-XL-2-512 (Peebles & Xie, 2022)), we generate images with random labels from the ImageNet dataset. For GANs, we train and evaluate with uniformly sampled latent codes. The default target generative model is SDXL-Turbo, for its state-of-the-art utility/efficiency balance.

Baselines. We compare with Stable Signature (Fernandez et al., 2023), WOUAF(Kim et al., 2023), and post-generation watermarking methods (Zhu et al., 2018; Tancik et al., 2019). To conduct a fair comparison, if not specified, we adopt the identical HiDDeN (Zhu et al., 2018) extractor from Stable Signature, which extracts a 48-bit message and is unbiased (i.e., the output bits are i.i.d. Bernoulli distributed on vanilla images, more details in Appendix B.3).

Hyperparameters. We train the mapping model in 2×10^4 steps with the batch size of 4. The optimizer is AdamW (Loshchilov & Hutter, 2017) with learning rate 1×10^{-4}. The bit length of user-specific watermarking message k is 48. See Appendix B.5 for more details.

Metrics. We measure efficacy of watermarks by the accuracy of attributing a watermarked image to the correct user (after correctly decided if the image is AI-generated) when the overall FPR$(\tau, \#user) = 10^{-6}$ (abbreviated as AttributionAcc@1E-6FPR), where a decision threshold τ is set to achieve the target false positive rate (see Appendix B.2 for details). By default, we evaluate on a dataset of 100,000 images, where $N = 1000$ users are assigned a unique random-sampled watermarking message and generates 100 images each. We measure the fidelity of a watermarked generator by Fréchet Inception Distance (FID) (Heusel et al., 2017), Structural Similarity Score (SSIM) (Wang et al., 2004) and Peak Signal-to-Noise Ratio (PSNR) with the original generator.

Table 1. **Generalizability** across four generative model architectures, as well as **comparable fidelity and robustness** to the baselines on SDXL-Turbo. For FID, we show in (color) the difference with regards to original. The FID of DiT-XL-2-512 and StyleGAN-XL are evaluated with their training dataset ImageNet (see Appendix B for details). (†): We mix generated images from the 4 models and then perform evaluation, demonstrating our method's unique capability of unified cross-model attribution.

		PSNR / SSIM ↑	FID ↓	AttributionAcc@1E-4FPR ↑ under attack: None	Crop	Brigh.	JPEG	Comb.
Methods	Dct-Dwt(Cox et al., 2007)	39.5 / 0.97	20.6 (−0.3)	0.62	0.00	0.00	0.00	0.00
	HiDDeN(Zhu et al., 2018)	32.0 / 0.88	20.7 (−0.2)	0.99	0.98	0.99	0.94	0.93
	WOUAF(Kim et al., 2023)	29.1 / 0.86	20.8 (−0.1)	0.99	0.99	0.99	0.92	0.91
	Stable Signature(Fernandez et al., 2023)	30.1 / 0.91	20.7 (−0.2)	0.99	0.99	0.99	0.93	0.85
	Unimark (Ours)	32.4 / 0.93	20.6 (−0.3)	0.99	0.99	0.99	0.90	0.93
Models	SDXL-Turbo (LDM)	32.4 / 0.93	20.6 (−0.3)	0.99	0.99	0.99	0.90	0.93
	LCM-XL (LCM)	32.3 / 0.91	15.0 (−0.6)	0.99	0.99	0.99	0.85	0.90
	DiT-XL-2-512 (DiT)	30.4 / 0.91	20.8 (−0.9)	0.99	0.94	0.87	0.60	0.50
	StyleGAN-XL (GAN)	31.1 / 0.94	3.36 (+0.1)	0.99	0.99	0.99	0.94	0.91
	Mixed †	34.2 / 0.96	-	0.99	0.99	0.96	0.88	0.84

Table 2. **Scalability** on users (U) and generative models (M). We assume each user generates an average of 100 images, each utilizing one of the possible generative models (e.g., 100M means 100 models to choose from).

Embedding Time	1M 1U	1M 10^2U	1M 10^4U	1M 10^6U	100M 10^6U
HiDDeN	**11 s**	**0.31 h**	31 h	3055 h	3055 h
StableSig	60 s	1.67 h	167 h	16667 h	16667 h
WOUAF	85.2 h	85.2 h	85.2 h	85.2 h	8521 h
Unimark	1 h	1 h	**1 h**	**1 h**	**100 h**

Extraction Time	1U 1M	1U 10^2M	100U 1M	100U 10^2M	100U 10^4M
WOUAF	30s	0.83h	0.83h	83h	8300h
HiDDeN/StableSig/ Unimark	**15s**	**15s**	**0.42h**	**0.42h**	**0.42h**

4.2. Generalizability

Table 1 reports fidelity metrics and the attribution accuracy under different transformations. Except evaluation on each model, we also conduct a mixed test, where each user chooses a random generative model to generate images. Our method generalizes well across the 4 different sota models, achieving 99% attribution accuracy for each model and the mixed test when not under attack or just under cropping.

4.3. Practical Scalability and Efficiency

Table 2 shows the total embedding and extraction time of each methods with a increasing number of users and generative models. Our training for SDXL-Turbo costs only 1 H100 GPU hour without additional optimization, resulting in the most scalable method in terms of the number of users and generative models. Besides, we utilize a universal extractor rather than training a extractor for each model like WOUAF, which is more scalable for extraction.

4.4. Communication Cost and Information Disclosure

Communication Cost. During the joint training, an image of size 256x256 and at least one encoded perturbation ψ_i as well as their gradients are transmitted every step between the provider and the regulator, which, in our experiments, results in the average transmission amount of 13.6MiB every step and totally 265.6GiB in 2×10^4 steps to achieve a satisfactory watermarking performance. In our simulation experiment of two processes communicating with each other through an intranet capped at 1000Mbps, the runtime overhead is approximately 34.8%. We anticipate that techniques like data compression before transmission could optimize the communication cost. Moreover, implementing a parallel strategy similar to Pipeline Parallel (Harlap et al., 2018) may significantly mitigate the associated runtime overhead.

Information Disclosure. To assess the risk of leaking θ, we propose a straightforward attack: estimate the magnitude of the original model weights W_0 from the magnitude of the encoded perturbation ψ_i. To facilitate quantitative analysis, we relax the threat model, allowing the adversary to know the position of modifications in the original model corresponding to each scalar in the ψ_i (unless permuted). Suppose weights w_j correspond to the a set of scalars s_j in ψ_i. We measure the leakage by calculating the correlation (pearson correlation coefficient) between $||s_j||_1$ and $||w_j||_1$. The correlation decreases from approximately 0.10 to 0.03 when transitioning from absolute perturbation to relative perturbation. Furthermore, after applying permutation, the correlation approaches zero. These results suggest that our method contributes to reducing information leakage.

Algorithm 1 Gray-box Constrained Training with transmission details between inventor (P) & regulator (R). We denote the gradient function as $\nabla_{\cdot}(\cdot)$, the gradient of a w.r.t. b as g_b^a, transferring a to b as $a \Rightarrow b$.

Require: generative model $\mathcal{G}_\theta$, encoding function $\mathcal{F}_\theta$, bit length d_m, perceptual/message loss function L_p/L_w, perceptual limit ϵ, perceptual weight λ, dataset $\mathbb{D}$, max projection iterations k, learning rate α.

P: $d_\psi \leftarrow |\mathcal{F}_\theta(\theta)|$ $\quad d_\psi \Rightarrow$ R
R: $\mathcal{M}_\Phi \leftarrow$ initialize(d_m, d_ψ)
while training not converged **do**
  P: $x \sim \mathbb{D}$
  P: $y \leftarrow \mathcal{G}_\theta(x)$
  R: $m_i \sim$ Binomial$(d_m, 0.5)$
  for $\iota = 1$ .. k **do**
    R: $\psi_i \leftarrow \mathcal{M}_\Phi(m_i)$ $\quad \psi_i \Rightarrow$ P
    P: $\theta_i \leftarrow \mathcal{F}_\theta^{-1}(\psi_i)$
    P: $y_i \leftarrow \mathcal{G}_{\theta_i}(x)$
    P: $\mathcal{L}_p \leftarrow L_p(y, y_i)$
    if $\mathcal{L}_p \leq \epsilon$: **break** // constrain L_p within ϵ
    P: $g_{\psi_i}^{\mathcal{L}_p} \leftarrow \nabla_{\psi_i}(\mathcal{L}_p)$ $\quad g_{\psi_i}^{\mathcal{L}_p} \Rightarrow$ R
    R: $g_\Phi^{\mathcal{L}_p} \leftarrow g_{\psi_i}^{\mathcal{L}_p} \cdot \nabla_\Phi(\psi_i)$
    R: $\Phi \leftarrow \Phi - \alpha \cdot$ Adam$(\Phi, \lambda \cdot g_\Phi^{\mathcal{L}_p})$
  end for $\quad y_i \Rightarrow$ R
  R: $g_{y_i}^{\mathcal{L}_w} \leftarrow \nabla_{y_i}(L_w(m_i, \mathcal{W}(y_i)))$ $\quad g_{y_i}^{\mathcal{L}_w} \Rightarrow$ P
  P: $g_{\psi_i}^{\mathcal{L}} \leftarrow (g_{y_i}^{\mathcal{L}_w} + \lambda \cdot \nabla_{y_i}(\mathcal{L}_p)) \cdot \nabla_{\psi_i}(y_i)$ $\quad g_{\psi_i}^{\mathcal{L}} \Rightarrow$ R
  R: $g_\Phi^{\mathcal{L}} \leftarrow g_{\psi_i}^{\mathcal{L}} \cdot \nabla_\Phi(\psi_i)$
  R: $\Phi \leftarrow \Phi - \alpha \cdot$ Adam$(\Phi, g_\Phi^{\mathcal{L}})$
end while

Ensure: $\mathcal{M}_\Phi$

4.5. Maintaining Fidelity and Robustness

Note that, similar to Stable Signature, our robustness is derived from the pre-trained robust extractor, which should be the same given the same extractor. We evaluate the robustness of our watermarking method against various attacks, including image-level attacks and model-level attacks.

Image-level attacks. We note that if the extractor is compromised (violating the threat model assumptions), an adversary can effectively remove or overwrite the watermark with a certain query budget to the extractor, as demonstrated in previous research (Jiang et al., 2023; An et al., 2024). We show the robustness of the watermark to various conventional image transformations such as cropping in Table 1. We refer to Appendix B.4 for details of the transformations. The results demonstrate the watermark's robustness across multiple generative models and various transformations. In the majority of experiments, attribution accuracy exceeded 0.9. Notably, this robustness was achieved without the necessity of involving transformations during the training phase of the mapping model, illustrating the efficacy of the pre-trained robust watermark extractor.

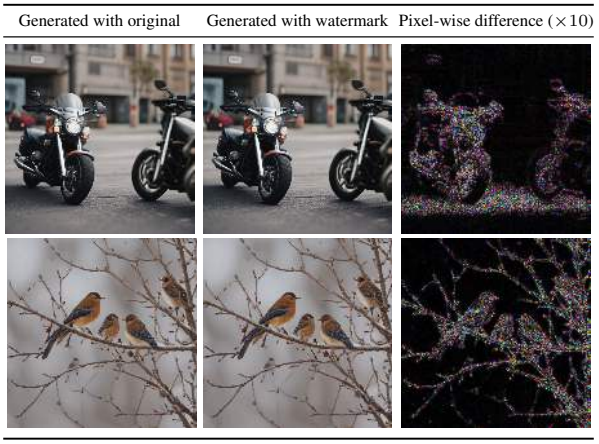

Figure 4. Qualitative results of images generated with watermarked SDXL-Turbo. More examples in App. D.

5. Limitations and Future Work

Limitations. Our watermarking operates at a *non-semantic* level, unlike semantic watermarking methods such as Tree-Ring (Wen et al., 2023). This represents a trade-off: while semantic watermarking could offer greater robustness, it comes at a prohibitively high cost. The extractor has to extract high-level semantic features for each image, e.g. via the whole reverse diffusion process in Tree-Ring. The computational demands and complexity make it impractical for large-scale deployment in most real-world scenarios. Still, we anticipate that future work could interpolate between the two "extremes" and find a better solution on balancing efficiency and robustness. Furthermore, we hypothesize a trilemma of watermarking in Appendix C.

Future Work. Future work can refine each component of our framework, including the universal extractor, mapping model architecture, perturbation encoding, and the training algorithm. Currently, these modules are implemented for demonstration purposes and present opportunities for further optimization and enhancement.

6. Conclusion

We introduce Unimark, a unified scalable watermarking framework for vision generative models, addressing the urgent necessity for user-wise attribution of AI-generated images. In addition to enhancing generalizability and scalability, we propose solutions to address novel real-world challenges of reducing inter-party communication costs and information disclosure. By extensive experiments, we demonstrate that our approach improves generalizability and scalability over existing methods while maintaining

state-of-the-art fidelity and robustness. We hope this work could facilitate future research on the regulation of synthetic content.

References

Ahmadi, M., Norouzi, A., Karimi, N., Samavi, S., and Emami, A. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146:113157, 2020.

An, B., Ding, M., Rabbani, T., Agrawal, A., Xu, Y., Deng, C., Zhu, S., Mohamed, A., Wen, Y., Goldstein, T., and Huang, F. Waves: Benchmarking the robustness of image watermarks. *ArXiv*, abs/2401.08573, 2024. URL https://api.semanticscholar.org/CorpusID:267028170.

Cox, I., Miller, M., Bloom, J., Fridrich, J., and Kalker, T. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2007. ISBN 9780080555805.

Czolbe, S., Krause, O., Cox, I., and Igel, C. A loss function for generative neural networks based on watson's perceptual model. *ArXiv*, abs/2006.15057, 2020. URL https://api.semanticscholar.org/CorpusID:220128042.

Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., and Douze, M. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3054–3058. IEEE, 2022.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., and Furon, T. The stable signature: Rooting watermarks in latent diffusion models. *ICCV*, 2023.

Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., and Holz, T. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pp. 3247–3258. PMLR, 2020.

Google DeepMind. SynthID. https://deepmind.google/technologies/synthid/. URL https://deepmind.google/technologies/synthid/.

Ha, D., Dai, A. M., and Le, Q. V. Hypernetworks. *ArXiv*, abs/1609.09106, 2016. URL https://api.semanticscholar.org/CorpusID:208981547.

Harlap, A., Narayanan, D., Phanishayee, A., Seshadri, V., Devanur, N. R., Ganger, G. R., and Gibbons, P. B. Pipedream: Fast and efficient pipeline parallel dnn training. *ArXiv*, abs/1806.03377, 2018. URL https://api.semanticscholar.org/CorpusID:47016772.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Hsu, C.-T. and Wu, J.-L. Hidden digital watermarks in images. *IEEE Transactions on image processing*, 8(1): 58–68, 1999.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021a.

Hu, S., Li, Y., and Lyu, S. Exposing gan-generated faces using inconsistent corneal specular highlights. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2500–2504. IEEE, 2021b.

Jeong, Y., Kim, D., Min, S., Joe, S., Gwon, Y., and Choi, J. Bihpf: Bilateral high-pass filters for robust deepfake detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 48–57, 2022.

Jiang, Z., Zhang, J., and Gong, N. Z. Evading watermark based detection of ai-generated content. *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023. URL https://api.semanticscholar.org/CorpusID:258557682.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

Kim, C., Min, K., Patel, M., Cheng, S., and Yang, Y. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. *ArXiv*, abs/2306.04744, 2023. URL https://api.semanticscholar.org/CorpusID:259108471.

Kishore, V., Chen, X., Wang, Y., Li, B., and Weinberger, K. Q. Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations*, 2021.

Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., and Guo, B. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5001–5010, 2020.

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. URL https://api.semanticscholar.org/CorpusID:14113767.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101, 2017. URL https://api.semanticscholar.org/CorpusID:3312944.

Lukas, N. and Kerschbaum, F. Ptw: Pivotal tuning watermarking for pre-trained image generators. 2023. URL https://arxiv.org/pdf/2304.07361.pdf.

Luo, S., Tan, Y., Huang, L., Li, J., and Zhao, H. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Luo, X., Zhan, R., Chang, H., Yang, F., and Milanfar, P. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13548–13557, 2020.

Marra, F., Gragnaniello, D., Verdoliva, L., and Poggi, G. Do gans leave artificial fingerprints? In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 506–511. IEEE, 2019.

Midjourney. Midjourney Terms of Service. https://docs.midjourney.com/docs/terms-of-service. URL https://docs.midjourney.com/docs/terms-of-service.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Nie, G., Kim, C. S., Yang, Y., and Ren, Y. Attributing image generative models using latent fingerprints. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:258212639.

OpenAI. Understanding the source of what we see and hear online. https://openai.com/index/understanding-the-source-of-what-we-see-and-hear-online/, a. URL https://openai.com/index/understanding-the-source-of-what-we-see-and-hear-online/.

OpenAI. Usage policies. https://openai.com/policies/usage-policies/, b. URL https://openai.com/policies/usage-policies/.

Peebles, W. S. and Xie, S. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2022. URL https://api.semanticscholar.org/CorpusID:254854389.

Qian, Y., Yin, G., Sheng, L., Chen, Z., and Shao, J. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pp. 86–103. Springer, 2020.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Nießner, M. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.

Sauer, A., Schwarz, K., and Geiger, A. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Sauer, A., Lorenz, D., Blattmann, A., and Rombach, R. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Seetharaman, D. and Barnum, M. There's a Tool to Catch Students Cheating With ChatGPT. OpenAI Hasn't Released It. https://www.wsj.com/tech/ai/openai-tool-chatgpt-cheating-writing-135b755a, 2024. URL https://www.wsj.com/tech/ai/openai-tool-chatgpt-cheating-writing-135b755a.

Si, C., Wang, X., Yang, X., Xu, Z., Li, Q., Dai, J., Qiao, Y., Yang, X., and Shen, W. FLoRA: Low-Rank Core Space for N-dimension. URL http://arxiv.org/abs/2405.14739.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Tancik, M., Mildenhall, B., and Ng, R. Stegastamp: Invisible hyperlinks in physical photographs.

2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2114–2123, 2019. URL https://api.semanticscholar.org/CorpusID:131773730.

The White House. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence. URL https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intel

Vukotić, V., Chappelier, V., and Furon, T. Are deep neural networks good for blind image watermarking? In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7. IEEE, 2018.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wen, Y., Kirchenbauer, J., Geiping, J., and Goldstein, T. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Wu, X., Xie, Z., Gao, Y., and Xiao, Y. Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 2952–2956. IEEE, 2020.

Yu, N., Skripniuk, V., Chen, D., Davis, L. S., and Fritz, M. Responsible disclosure of generative models using scalable fingerprinting. *ArXiv*, abs/2012.08726, 2020. URL https://api.semanticscholar.org/CorpusID:229212743.

Yu, N., Skripniuk, V., Abdelnabi, S., and Fritz, M. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 14448–14457, 2021.

Zhang, C., Benz, P., Karjauv, A., Sun, G., and Kweon, I. S. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 33:10223–10234, 2020.

Zhang, K. A., Cuesta-Infante, A., Xu, L., and Veeramachaneni, K. Steganogan: High capacity image steganography with gans. *ArXiv*, abs/1901.03892, 2019. URL https://api.semanticscholar.org/CorpusID:59413797.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 657–672, 2018.

Ethics & Reproducibility Statement

We note that the framework is immature for widespread deployment due to the ethical concerns of potential system abuse. It currently lacks rigorous cryptographic and algorithmic guarantees to ensure the regulator *credibly commit* to a fixed set of rules according to which generated content will be attributed. Mere informal commitments to maintain proper practices are insufficient. Future work should establish formal verification mechanisms and provable security properties to ensure system integrity and user trust. Our code and model checkpoints will be publicly released upon publication.

A. Additional Related Works

In this section, we discuss related works of watermarking methods that attribute AI-generated images to their responsible users (user-wise attribution methods). Based on where the watermarking message is *introduced* to the image, we categorize existing methods into three types: *post-generation*, *pre-generation* and *in-generation* watermarking. Note that all three types of watermarks end up embeded in *all* generated images to achieve user-wise attribution. For comparison, to achieve model intellectual property protection, the watermark can be triggered to be embeded in only *some* generated images (e.g., via backdoor).

Post-generation watermarking methods (Hsu & Wu, 1999; Zhu et al., 2018; Zhang et al., 2020; 2019; Vukotić et al., 2018; Ahmadi et al., 2020; Fernandez et al., 2022; Kishore et al., 2021; Luo et al., 2020) watermark the images after they are generated. HiDDeN (Zhu et al., 2018), a classic end-to-end learning-based method, embeds invisible watermark into the given image. It adopts a message encoder to encode messages into images and a message decoder to extract messages embedded in the encoded images. Furthermore, it employs adversarial training to enhance robustness against common image processing operations. Unfortuantely, all post-generation methods (i) suffer from low scalability since the computational overhead grows with the number and size of gernerated content, and (ii) are easy to remove from the source code in scenarios like open-source generative models.

Pre-generation watermarking methods watermark the images at the input stage. Nie et al. (Nie et al., 2023) propose embedding a unique pattern into the latent codes; however, their method is limited to support only GANs. For LDMs, Tree-Ring Watermark (Wen et al., 2023) adds pre-defined keys to the initial noise of diffusion models, which is then used to generate an image. It recovers the noise and verifies the key robustly by perfroming reversed diffusion process on the generated image. However, it relies on access to the diffusion model and an expensive reversed diffusion process for each image, rendering it unsuitable for practical deployment.

In-generation watermarking methods (Yu et al., 2021; 2020; Lukas & Kerschbaum, 2023; Fernandez et al., 2023; Kim et al., 2023) merge the generation process and watermarking process via *model watermarking*. Images generated by a watermarked model are born with watermarks, which is more efficient and scalable. Yu et al. (Yu et al., 2021) propose modifying training data to watermark the model. A series of works then employ modulation, a technique used in style-based generators (Karras et al., 2019; 2021; Sauer et al., 2022), to watermark the model by scaling convolutional filters (Yu et al., 2020; Kim et al., 2023) or kernels (Lukas & Kerschbaum, 2023). Yu et al. (Yu et al., 2020) train GANs from scratch with modulation as watermarking. PTW (Lukas & Kerschbaum, 2023) applies modulation on pretrained GANs to improve efficiency. WOUAF (Kim et al., 2023) migrate this idea to LDMs. However, these methods require training a separate extractor for each model. Stable Signature (Fernandez et al., 2023) proposes to watermark LDMs (and subsequently images generated by it) by fine-tuning the model against a pre-trained watermark decoder with a specific watermarking message for each user. They levarage a pre-trained watermark decoder to fine-tune the VAE decoder of latent diffusion models, forcing the VAE decoder to output images born with watermarks. However, its scalability is limited. Its computational demand increase linearly with the number of users since it necessitates a fixed-time fine-tuning process for each individual. In contrast, our method achieves fixed training costs regardless of user growth.

B. Experimental Details & Hyperparameters

B.1. Threat Model

In this section, we describe our threat model in detail.

1) Adversary's Objective: The adversary (malicious user) aims to remove the watermark from the watermarked images without sacrificing the image quality or the model utility.

2) Adversary's Capabilities: We assume the watermark extractor is private and a malicious user has no access to detection/attribution API of the regulator (**no-box**). Such threat model arises when the regulator restricts the access of its detection/attribution API to a small set of trusted customers, e.g., Google's SynthID (Google DeepMind) adopts this threat model.

Based on this, our framework supports two scenarios:

i) API-Scenario: The model inventor trains and hosts the model, then offers watermarked API services to users (e.g., like Midjourney). The *usage policy* constrains users from generating disapproved images with this service. In this scenario, the adversary can manipulate the watermarked images y_i to remove the watermark (**image-level attack**).

ii) Distribution-Scenario: The model inventor trains the model, embeds the watermark, and subsequently distributes the watermarked models to users for local deployment. This could be an open-source free download (e.g., via HuggingFace) or an one-time purchase. The *usage policy* requires each user to keep its own copy confidential and take responsibility for the generated images. In this scenario, beside image-level attacks, the adversary can also manipulate the watermarked model $\mathcal{G}_{\theta_i}$ to remove the watermark in its generated images (**model-level attack**).

B.2. Statistical Test

1) Detection: Following the practice of Stable Signature (Fernandez et al., 2023), we extract the message m' from an image y and compare it to m_i to test the statistical hypothesis H_1: "y was generated by u_i". Following (Fernandez et al., 2023), we *flag* the image if the matching bits between m_i and m'_i meets or exceeds a threshold $\tau \in \{0, \dots, d_m\}$. We adjust τ to balance the True Positive Rate (TPR) (i.e., the probability of flagging an image generated by u_i) and the False Positive Rate (FPR) (i.e., the probability of flagging a vanilla image). For instance, when $\tau = 0$, we flag all images so $\text{FPR}(\tau) = \text{TPR}(\tau) = 1$. The TPR is directly measured, while the FPR is inferred from the CDF of a binomial distribution with parameters $(d_m, 0.5)$.

2) Attribution: Given N users (across various generative models), the attribution process involves testing N distinct detection hypotheses. Therefore, the overall $\text{FPR}(\tau, N) = 1 - (1 - \text{FPR}(\tau))^N \approx N \cdot \text{FPR}(\tau)$. If all hypotheses are rejected, we conclude that the image was not generated by any of the N users. Otherwise, we attribute to $\text{argmin}_i L_w(m_i, m')$ as the responsible user. In practice, we adjust τ to achieve a target overall $\text{FPR}(\tau, N)$, e.g. 10^{-4}.

B.3. Evaluated Gernerative Models

We experiment with 4 SOTA generative models representing 4 different model architectures: Latent Diffusion Model (LDM), Latent Consistency Model (LCM), Diffusion Transformer (DiT), and Generative Adversarial Network (GAN). The models evaluated are SDXL-Turbo (Sauer et al., 2023), LCM-XL (Luo et al., 2023), DiT-XL-2-512 (Peebles & Xie, 2022), and StyleGAN-XL (Rombach et al., 2022), respectively. The details of these models are provided below.

1) SDXL-Turbo (Sauer et al., 2023): a distilled version of SDXL (a state-of-the-art latent diffusion model open-sourced by StabilityAI). SDXL-Turbo is based on a training method called Adversarial Diffusion Distillation (ADD) (Sauer et al., 2023), which allows sampling large-scale foundational image diffusion models in 1 to 4 steps at high image quality. We conduct the evaluation on the preferred resolution 512×512 with 1 inference steps.

2) LCM-XL (Luo et al., 2023): another distilled version of SDXL 1.0, where only the unet is replaced. Viewing the guided reverse diffusion process as solving an augmented probability flow ODE (PF-ODE), LCMs predict its solution directly in latent space, achieving fast inference in 2 to 4 steps. We conduct the evaluation on the preferred resolution 768×768 with 4 inference steps.

3) DiT-XL-2-512 (Peebles & Xie, 2022): the diffusion transformer open-sourced by facebook. We conduct the evaluation on the its preferred resolution 512×512 with 25 inference steps and label-to-image tasks.

4) StyleGAN-XL (Sauer et al., 2022): one of the latest Generative Adversarial Network (GAN). We conduct the evaluation on the variant trained on ImageNet with resolution 512×512 .

B.4. Image Transformations

For the image transformations, we adopt the same benchmarks as in StableSignature (Fernandez et al., 2023). If not specified, for cropping, we randomly crop the image to remain only 10% fraction; for brightness, we randomly adjust the brightness

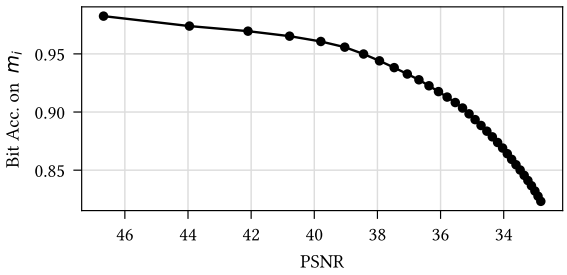

Figure 5. Model purification results. The PSNR is calculated between the images generated by the purified model and the watermarked images.

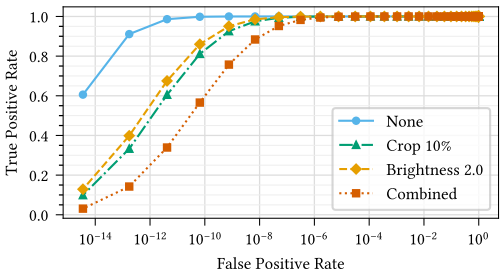

Figure 6. Detection TPR/FPR curve under different transformations on SDXL-Turbo.

by a factor of 2.0; for JPEG compression, we compress the image with a quality factor of 80. The combined transformation is a combination of Crop 50%, Brightness 1.5 and JPEG 80.

B.5. Other Watermarking Methods

For Stable Signature (Fernandez et al., 2023), we use the hyperparameters specified in the original paper: #total steps = 100; #warmup steps = 20; batch size = 4; optimizer = AdamW; learning rate = 5×10^{-4}; $\lambda_w = 1$; $\lambda_i = 0.2$; bit length = 48. For WOUAF (Kim et al., 2023), the hyperparameters are as following (as specified in the original paper, except for an increase in bit length from 32 to 48 and a migration from SD2-1 to SDXL-Turbo to ensure fair comparability): #total steps = 50000; #GPUs = 4; batch size = 2 (for each GPU); optimizer = AdamW; learning rate = 10^{-4}; $\lambda_w = 1$; $\lambda_i = 1$; bit length = 48; gradient accumulation steps = 8. Same as the original implementation, we scale learning rate with #GPU $\times$ batch size $\times$ gradient accumulation steps.

C. The Watermark Trilemma Hypothesis

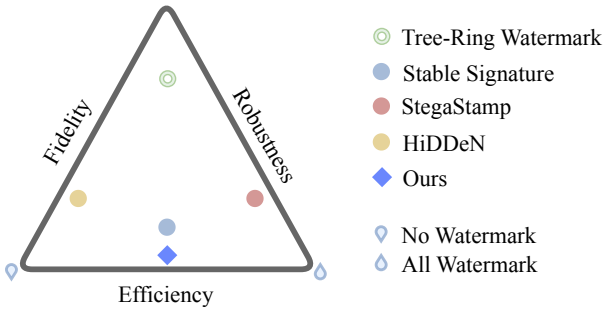

Figure 7. **The Watermark Trilemma Hypothesis.** We hypothesize the trilemma exists in the watermarking system space (each point is a watermarking system). A system's Efficiency/Robustness/Fidelity is denoted as its distance to each edge (the closer the better). Thus the three vertices represent achieving optimal ER, EF, and RF respectively. But no system can achieve optimal ERF at the same time (trilemma).

Figure 7 shows the efficiency/robustness/fidelity trilemma in watermarking literature with all milestone works in it. In this model, robustness encompasses accuracy (or efficacy), as it is primarily evaluated based on accuracy metrics under diverse attack scenarios, thereby serving as a comprehensive measure of accuracy. And fidelity (or quality) includes stealthiness, because the two goals are often aligned - an invisible watermark is both stealthy and faithful to the original image. Efficiency is defined as the system's capacity to handle an increasing number of inventors/models/users/images. We hypothesize that any watermarking system is a trade-off between efficiency, robustness, and fidelity (i.e., trilemma).

To enhance the comprehensiveness of this analytical framework, we propose two methods that could correspond to two of the three vertices. No Watermark (EF): The system does not embed any watermark, thus achieving the highest fidelity and efficiency, but no robustness. All Watermark (ER): The system utilizes the entirety of the content for watermarks, thereby achieving maximum robustness and efficiency, but no fidelity. The third vertex (FR) could correspond to a computationally infeasible method with least efficiency (e.g., requiring the ground-truth perceptual distance metric, visual perception of all humanity).

D. Additional Qualitative Results

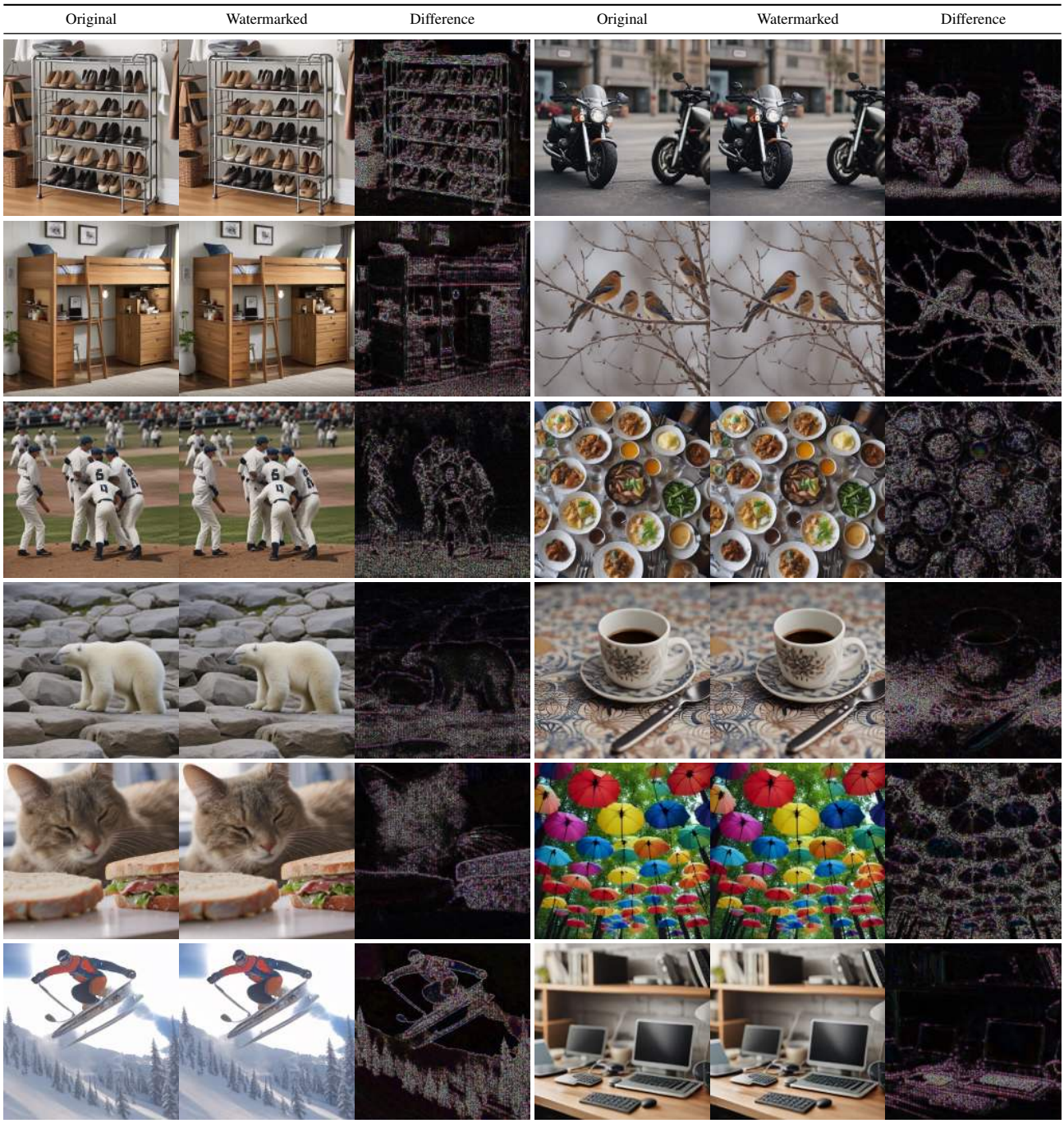

Figure 8. Qualitative results of SDXL-Turbo on prompts of the validation set of MS-COCO, at resolution 512 and 1 inference step.

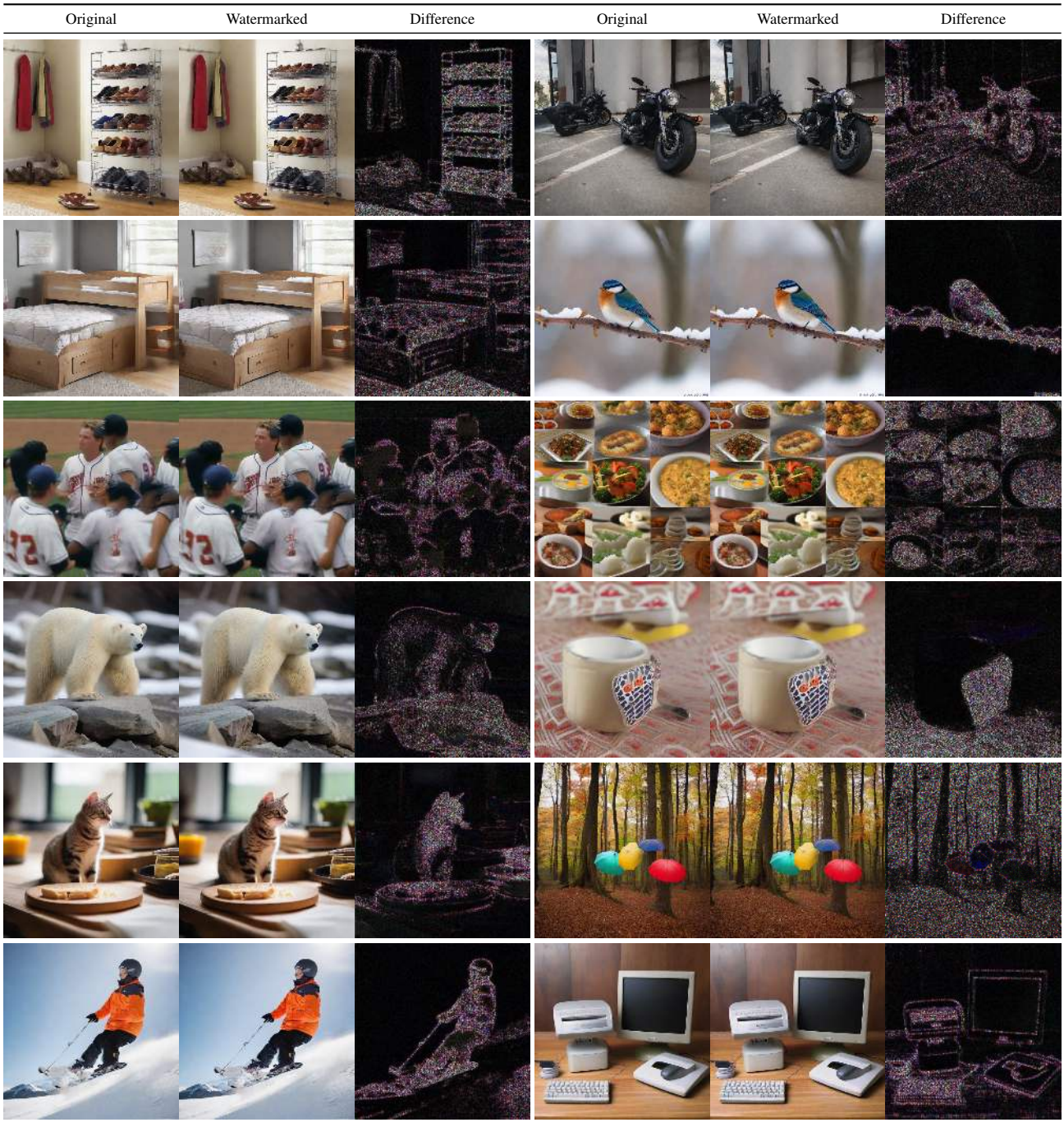

Figure 9. Qualitative results of LCM-XL on prompts of the validation set of MS-COCO, at resolution 768, guidance scale 8 and 4 inference steps.

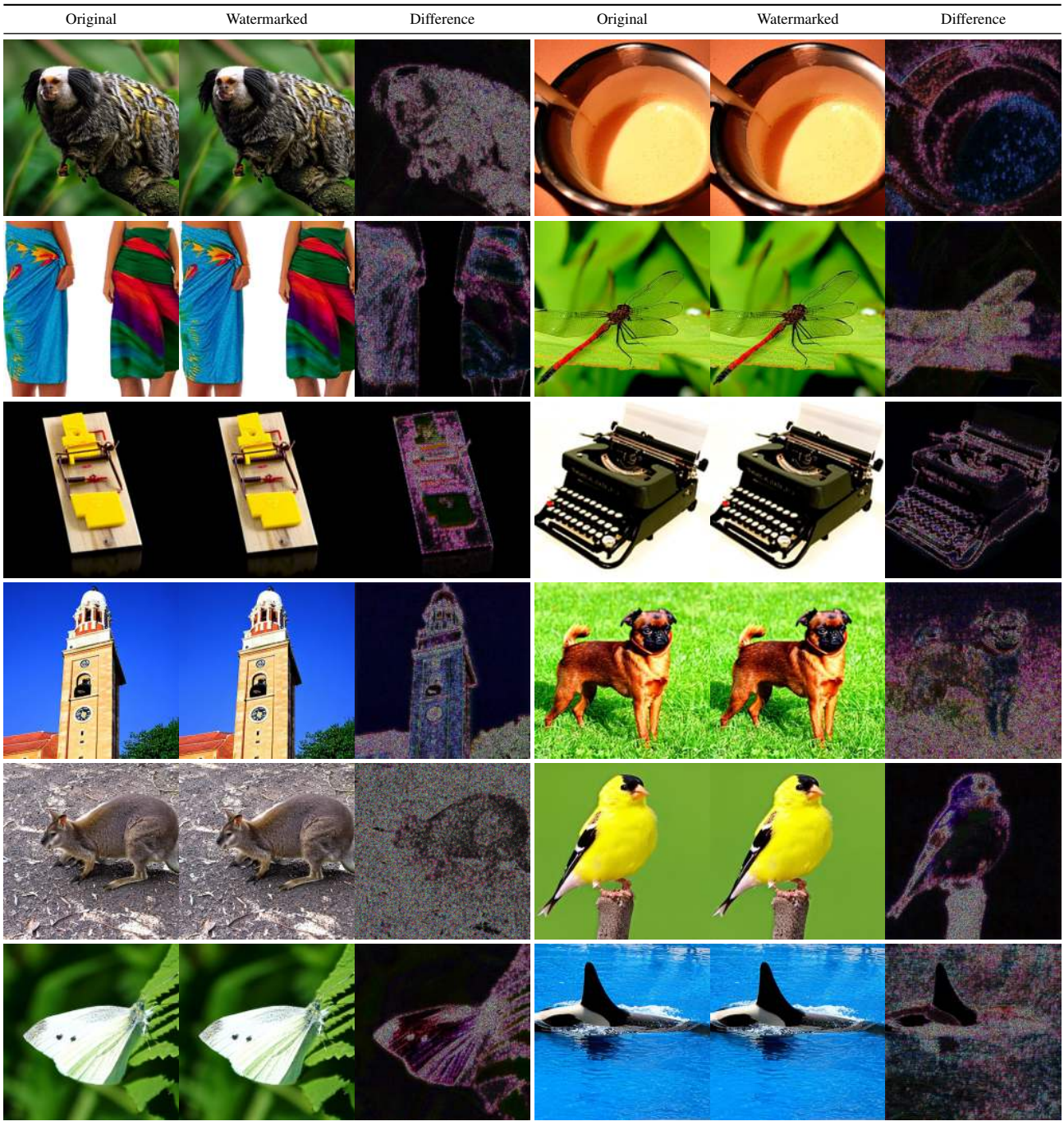

Figure 10. Qualitative results of DiT-XL-2-512 on random class label of ImageNet, at resolution 512 and 25 inference steps.

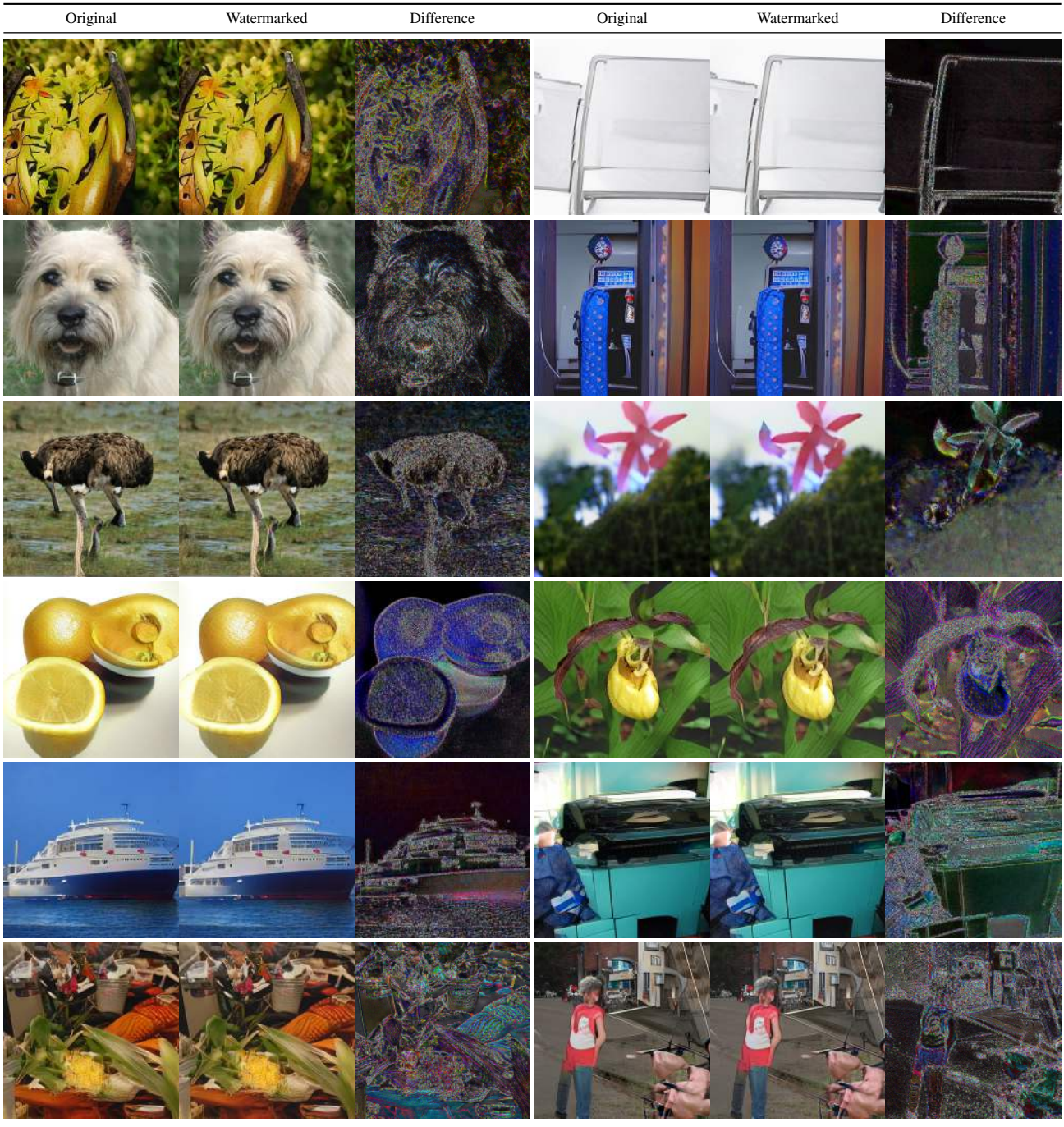

Figure 11. Qualitative results of StyleGAN-XL on random latent code at resolution 512.