TRACING THE MISUSE OF PERSONALIZED TEXTUAL EMBEDDINGS FOR TEXT-TO-IMAGE MODELS

Weitao Feng[1], Jiyan He[1], Jie Zhang[3]*, Tianyi Wei[2], Wenbo Zhou[1], Qing Guo[3], Weiming Zhang[1], Tianwei Zhang[2], Nenghai Yu[1]
[1]University of Science and Technology of China, [2] Nanyang Technological University,
[3]Centre for Frontier AI Research, Agency for Science, Technology and Research (A*STAR)

ABSTRACT

Text-to-Image (T2I) models have achieved great success in generating high-quality images with diverse prompts. The emerging personalized textual embedding technology further empowers T2I models to create realistic images based on users' personalized concepts. This leads to a new AI business, with many commercial platforms for sharing or selling valuable personalized embeddings. However, this powerful technology comes with potential risks. Malicious users might exploit personalized textual embeddings to generate illegal content. To address this concern, these public platforms need reliable methods to trace and hold bad actors accountable. In this paper, we introduce *concept watermarking*, a novel approach that embeds robust watermarks into images generated from personalized embeddings. Specifically, an encoder embeds watermarks in the embedding space, while a decoder extracts these watermarks from generated images. We also develop a novel end-to-end training strategy that breaks down the diffusion model's sampling process to ensure effective watermarking. Extensive experiments demonstrate that our *concept watermarking* is effective for guarding personalized textual embeddings while guaranteeing their utility in terms of both visual fidelity and textual editability. More importantly, because the watermark exists at the concept level, it is robust against different processing distortions, diffusion sampling configurations, and adaptive attacks. Ablation studies are also conducted to validate the design rationale of each key component.

1 INTRODUCTION

With the rapid progress of generative models and cross-modal visual and language representation learning (Radford et al., 2021; Weng et al., 2023), it becomes remarkably simple to create high-resolution and realistic images from natural language descriptions (Dhariwal & Nichol, 2021; Nichol et al., 2022; Wei et al., 2022; 2023; Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022). We have witnessed the emergence of diverse Text-to-Image (T2I) generative models, represented by the diffusion model (Ho et al., 2020) and its variants. Although powerful, these T2I models have limited capabilities to handle the generation tasks with update-to-date or personalized concepts. Examples of such concepts include an ordinary person in our daily life, a unique object, or an artistic style that has never been seen before (Richardson et al., 2023). It is urgent to grant the T2I models the capability of generating arbitrary concepts.

To achieve this, researchers proposed some lightweight personalization techniques to customize T2I models (Gal et al., 2022; Ruiz et al., 2023; Kumari et al., 2023; Gal et al., 2023; Shi et al., 2023). One of the most popular solutions is Textual Inversion (TI) (Gal et al., 2022), which trains a personalized textual embedding for one specific concept. This textual embedding can be seamlessly integrated into a T2I model to generate images of this concept. Figure 1 (a) shows an example of such technique. The Stable Diffusion (SD) model is unable to generate a correct image of the "Toronto Tower" because its training dataset does not provide a sufficient impression of it. To address this, we can prepare a few images of the Toronto Tower and use them to train the personalized

[*] Corresponding author.

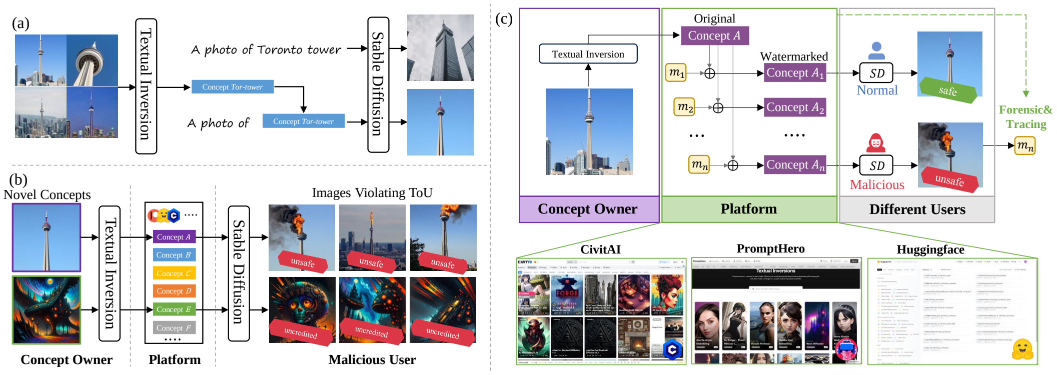

Figure 1: (a) Textual Inversion empowers Stable Diffusion to produce more realistic images via personalized textual embeddings (highlighted in blue). (b) The concept sharing scenario and its potential misuse, e.g., generating unsafe images and inducing copyright infringement. (c) The overview of *concept watermarking* for tracing the misuse of personalized concept (i.e., textual embedding).

textual embedding of the concept "Tor-tower." This textual embedding can then be added to the embedding dictionary of the SD model. Consequently, this enhanced SD model can generate images of the Toronto Tower using prompts like "A photo of Tor-tower."

The advent of textual embedding fosters a new AI business. As training textual embeddings require a certain level of expertise in data collection and hyper-parameter configuration, new commercial platforms emerge for the sharing or sale of various personalized concepts in the form of textual embeddings. For instance, CivitAI (Civitai) is a popular website for AI content creators to share their personalized concepts, with 3 million registered users. PromptHero (prompthero) is the top-1 website for prompt engineering, offering the sharing of personalized textual embeddings. HuggingFace (Huggingface), one of the most popular AI communities, provides a wide range of personalized concepts uploaded by users. Figure 1 (b) describes the common scenario of concept sharing, wherein there are three parties: 1) the *concept owner* owns some valuable personalized textual embeddings representing novel concepts; 2) the *concept-sharing platform* allows the concept owners to upload their valuable concepts for sharing or for profit; 3) the *concept user* can download/buy his desired concept and plug it into the off-the-shelf T2I model (e.g., SD model) for generating images with this new concept.

Problem Statement. Unfortunately, the personalized textual embedding technique and concept-sharing paradigm bring new security concerns. They can exacerbate the potential misuse of generative models, allowing malicious users to generate harmful or illegal content of more concepts with less effort. Figure 1 (b) shows two examples of such misuse. (1) A malicious user can download concepts representing landmarks to create dangerous fake images to cause public panic or concepts representing specific characters to produce violent, nude, or inappropriate images (not shown here due to ethical concerns). (2) A malicious user can leverage these concepts to generate images in similar styles without giving proper credit to the original authors, which may constitute copyright infringement (cri) and significantly affect the motivation of creative creators.

The concept-sharing platform is responsible for detecting the violations of the terms-of-use (ToU) and tracing the misuse of their concepts. A promising solution is to apply watermarking to personalized concepts. Adding watermarks to AI-generated content has increasingly become a consensus among large tech-companies and governments (OpenAI; Walker; Whitehouse). In our context, as shown in Figure 1 (c), the platform can integrate distinct watermark strings into the pristine concept embeddings to get different concept versions for different users. Subsequently, for any suspicious image on the Internet, the platform can apply a private decoder to extract the watermark from it. Based on the watermark information, it can trace the malicious user and provide evidence for forensics and accountability.

Challenges. However, it is non-trivial to achieve an effective concept watermarking. The Stable Diffusion official repository (Diffusion) lists some watermarking methods as options, e.g., DWT-DCT (Rahman, 2013), DWT-DCT-SVD (Rahman, 2013), and RivaGAN (Zhang et al., 2019). Researchers have also proposed more advanced watermarking solutions including StegaStamp (Tancik et al., 2020), CIN (Ma et al., 2022), and RoSteALS (Bui et al., 2023). However, these solutions are mainly used for the protection of conventional image media. When applied to the personalized textual embedding scenario, they all fail (as displayed in Table 1). This is mainly due to the unique features of textual embeddings and its extra requirements for *concept watermarking*:

1. *Fidelity*. For traditional image watermarking, fidelity means preserving the original visual quality of the watermarked images. However, *concept watermarking* has an extra fidelity requirement: preserving the generation ability and editability of the watermarked embeddings. Conventional techniques embed watermark patterns at the pixel level, which could easily get lost during the training process of personalized textual embedding.
2. *Robustness*. Prior image watermarking methods mainly consider the robustness against some digital post-processing distortions (e.g., affine transformations and JPEG) and cross-channel transmission distortions (e.g., printing and scree-shooting), which are all at the pixel space. However, *concept watermarking* faces more severe distortions such as the change of layouts and backgrounds guided by diverse prompts. Moreover, it is required to resist different diffusion processes with diverse sampler methods, configurations, model versions, etc.

Thus, it is important to design a new watermarking solution tailored to personalized embeddings.

Contributions. To address the above challenges, we present a new watermarking methodology with innovative designs of *training strategies*, *model architectures*, and *loss functions*. **First**, to balance the trade-off between fidelity and watermark extraction effectiveness, we adopt a progressive training strategy that initially achieves acceptable fidelity before refining for effectiveness. Also, considering the practical scenario where the concept-sharing platform has limited training images of the concept owner, we introduce a sampling-online training framework. This framework back-propagates gradients based on denoised results, strategically detaching gradients at suitable points to ensure computational efficiency (see Figure 2 and Sec. 4). **Second**, since model architecture significantly impacts fidelity (see Table 4), we employ the best U-Net with long-range skip connections as our watermark encoder. **Third**, besides pixel-level distortions, the malicious user may deliberately add distortion to the watermarked concept to remove the watermark. Therefore, we design a contrastive loss to improve the robustness against such embedding level distortions (See Table 6) to remedy this threat. Extensive experiments demonstrate that our proposed *concept watermarking* approach can effectively guard Textual Inversion against malicious usage. Specifically, our *concept watermarking* exhibits great robustness against different diffusion sampling configurations, such as samplers, sampling steps, CFG scales, and base models. It is also inherently robust against most pixel-level distortions as it is added within the highly compressed semantic space of the text encoder. Furthermore, it can resist some potential adaptive attacks. Many ablation studies are conducted to verify our key designs.

In summary, the primary contributions of our work are concluded as follows:

- We point out the necessity of tracing the misuse in the scenario of concept sharing and propose the novel *concept watermarking* as a feasible solution.
- To overcome the challenges for *concept watermarking*, we adopt a progressive training strategy to satisfy fidelity and explore different decoder architectures for effective forensics. To resist deliberate concept-level distortion, we design a contrastive loss.
- Extensive experiments demonstrate that our proposed method is effective for protecting textual embeddings comprehensively when facing various attacks. We also conduct many ablation studies to justify our elaboration.

2 Personalized Textual Embedding

The mainstream SD model has a fixed and finite knowledge base. It is unable to generate many unique and emerging concepts (e.g., Toronoto Tower in Figure 1 (a)), which appear less frequently in its training data. To adapt to users' diverse demands, it is an urgent requirement to enhance the model's knowledge with arbitrary concepts efficiently, an operation we typically refer to as *personalization*.

Researchers have proposed many personalization techniques, e.g., Textual Inversion (Gal et al., 2022), DreamBooth (Ruiz et al., 2023), Lora-DreamBooth (Hu et al., 2021). Among them, Textual

Inversion (TI) is the most attractive method. We only need to train a textual embedding for one specific concept, which can be seamlessly integrated into existing SD models to generate high-quality images with this new concept. TI offers several advantages over other personalization methods. (1) *Lightweight*: a textual embedding has a much smaller size, making it easy to share. For instance, a textual embedding for SD version 1.5 is approximately 30KB, whereas a personalized model fine-tuned using Dreambooth is more than 5 GB. (2) *User-friendly*: as a plug-in method, a user only needs to add the embedding to the embedding folder of the SD model, which can then generate the output with the desired concept. Due to these appealing features, we are witnessing many online platforms for users to share or sell their valuable concepts in the form of textual embeddings (Civitai; Huggingface; Patreon).

Technically, in order to create a textual embedding for a personalized concept with Textual Inversion (Gal et al., 2022), we need to prepare a few images with the target concept. Then, we optimize a textual embedding $\mathbf{s}^*$ based on a frozen SD model $\boldsymbol{\epsilon}_\theta$ by minimizing the distance between the generated and targeted images. This is formulated as follows:

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \mathbb{E}_{\mathbf{z}_0 \sim \mathcal{E}(\mathbf{x}), \mathbf{y}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t, t, \boldsymbol{\tau}_{\theta', \mathrm{s}}(\mathbf{y}) \right) \right\|_2^2 \right], \tag{1}$$

where $\mathbf{x}$ is one image from a set of the target images and $\mathbf{y}$ is the condition (i.e., different prompts), $\boldsymbol{\tau}$ is the text-encoder. This optimization process allows the textual embedding $\mathbf{s}^*$ to capture some detailed features of the target concept. Since $\mathbf{s}^*$ does not correspond to any existing vocabulary in any language, it is referred to as a pseudo-word embedding. In other words, the pseudo-word embedding $\mathbf{s}^*$ can be considered as the textual representation of the target concept.

We also provide preliminary of diffusion models in Appendix A.2

3 Related Work about Watermarking

Watermarking is a common solution for the copyright protection of multimedia. In the era of generative AI, this technology is applied to guard the AI-Generated Content. Specifically for T2I tasks, we can add watermarks on the generated images before releasing them to the public. These watermarks can be later extracted for multiple purposes: e.g., distinguishing if an image is AI-generated or real, identifying the harmful generated images, and tracing the malicious users, etc. Many large tech companies and government have adopted this proactive detection strategy (OpenAI; Walker; Whitehouse). Existing watermarking methods for AI-generated images can be categorized into two types[1]: (1) *post-hoc watermarking*: this strategy applies conventional methods to add watermarks on the images after they are generated. For example, the SD official repository (Diffusion) provides watermarking options with some suggested methods such as DWT-DCT (Rahman, 2013), DWT-DCT-SVD (Rahman, 2013), and RivaGAN (Zhang et al., 2019). (2) *Watermarking with diffusion*: this strategy combines the watermark embedding with the image generation (i.e., diffusion) process, achieved by modifying certain components of the diffusion models. For instance, we can adjust the diffusion sampling process to learn a specific watermarked sampling (Wen et al., 2023; Alemohammad et al., 2023), or embed watermarks into the Variational Autoencoder of the SD models (Fernandez et al., 2023). After that, images generated by these models will inherit the desired watermark automatically.

These methods may work well for the base SD models. However, when applied to personalized embeddings, they become completely ineffective. Firstly, these methods require full control over the sampling process when using the Stable Diffusion model, to add watermarks either during or post diffusion. This prerequisite is not met in our context, as concept users employ the concepts locally, so the concept sharing platform cannot add the watermarks. Secondly, utilizing image watermarking techniques to train Textual Inversion proves unsuccessful (see our evaluations in Sec. 5.2). This is mainly because they fail to satisfy the extra requirements introduced by the personalized process, where concepts are compressed into high-dimensional embedding spaces. To combat these, **for the first time, we introduce a novel concept watermarking approach, dedicated to the protection of personalized embeddings**.

[1] Note that in addition to protecting the AI-Generated Content, watermarking can also be used for the copyright protection of diffusion models (Zhao et al., 2023; Liu et al., 2023) and training datasets (Zhao et al., 2023; Cui et al., 2023). These methods have totally different goals from ours, and hence are not considered in this paper.

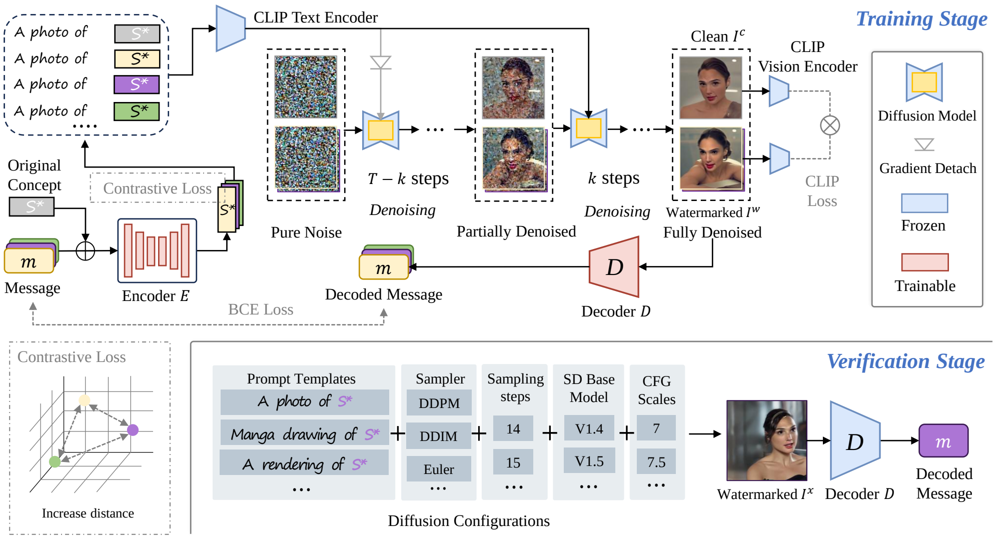

Figure 2: The overall framework of our proposed *concept watermarking*. $\oplus$ and $\otimes$ represent concatenate and cosine-similarity respectively. In the training stage, we jointly train the Encoder and Decoder to embed watermarks into the Textual Inversion embedding with online sampling, while ensuring the generation of semantically coherent images I^c and I^w. In the verification stage, we use different diffusion configurations, and the watermark can be extracted from the generated images I^x.

4 METHODOLOGY

4.1 OVERVIEW

The overall pipeline of our methodology is illustrated in Figure 2, which consists of two stages. In the training stage, the encoder and decoder are jointly trained. Specifically, for each iteration, a random bit string message $\mathbf{m}$ is first mapped to the hidden dimension size of concepts and then concatenated with the original concept, serving as the input for the encoder E. The encoder incorporates the watermark into the embedding, i.e., substituting the original embedding with the watermarked embedding for the corresponding tokenized prompts. The diffusion model accepts the encoded prompts with the original concept and watermarked concept as input, and generates both clean and watermarked images. The decoder D aims to successfully extract the corresponding embedded watermark. In the verification stage, we aim to extract the corresponding message from the generated images based on the watermarked concept, under different prompts and diffusion configurations.

4.2 TRAINING STAGE

4.2.1 WATERMARK EMBEDDING

This step is responsible for encoding the input bit string to the original Textual Inversion concept, meanwhile guaranteeing the utility of the watermarked concept.

Our watermark encoder, denoted as E, accepts the Textual Inversion concept $\mathbf{s} \in \mathbb{R}^{\kappa \times d_\tau}$ and the watermark message $\mathbf{m} \in \{0,1\}^q$ as its inputs. In order to match the size of the Textual Inversion embedding, we employ a linear layer to map the bit message $\mathbf{m}$ to the same size as the embedding size d_τ. Here, κ signifies the number of tokens in the concept, d_τ denotes the representation dimension of the text encoder, and q indicates the length of the watermark bit string. Following this, the mapped message and embedding are merged to form a representation of dimension $\mathbb{R}^{(\kappa+1) \times d_\tau}$. By treating $\kappa + 1$ as the channel, d_τ as the sequence length for 1D conv, the U-Net structure can be utilized to enable the effective encoding of the message within the Textual Inversion concept. We discover that long-range skip connections in U-Net can effectively control the distance between the watermarked and original Textual Inversion embeddings, resulting in high editability. More results with different architectures can be found in the ablation study in Sec. 5.4.

For training, we incorporate a regularization loss that calculates the L_2 distance between the original embedding $\mathbf{s}$ and the watermarked embedding $\mathbf{s^m} = E(\mathbf{s}, \mathbf{m})$:

$$\mathcal{L}_{\text{Reg}} = \|\mathbf{s} - E(\mathbf{s}, \mathbf{m})\|_2^2, \tag{2}$$

whose purpose is to facilitate the rapid convergence of the encoder output to the original embedding in the early training stage, enabling a favorable starting point and significantly accelerating the training process.

Additionally, to improve the embedding-level robustness, we devise a contrastive loss, which aims to increase the distance between Textual Inversion embeddings with different watermarks. Its calculation is similar to the contrastive learning (Chen et al., 2020) manner, but involves only negative pairs. The similarity is defined as the cosine values of two vectors. The formulation is as follows:

$$\mathbf{r}_i = \frac{E(\mathbf{s}, \mathbf{m}_i) - \mathbf{s}}{\|E(\mathbf{s}, \mathbf{m}_i) - \mathbf{s}\|_2}, \quad i = \{1, ..., N\}, \tag{3}$$

$$\mathcal{L}_{\text{Cst}} = \frac{1}{N(N-1)} \sum_{i,j=0, i\neq j}^{N} \langle \mathbf{r}_i, \mathbf{r}_j \rangle, \tag{4}$$

where N denotes the batch size and $\mathbf{m}_i$ refers to the i-th watermark message employed by the encoder. Incorporating this loss can increase the distance between different watermarked embeddings. In this way, when an attacker directly manipulates the embedding, it is less likely to fall into the decision boundary of another watermark message. Our ablation study in Sec. A.10 will demonstrate this.

4.2.2 Sampling Online Training

We select Stable Diffusion (Rombach et al., 2022) as our generative model. During training, we freeze the parameters of the diffusion model $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})$, where $\mathbf{y}$ represents the text condition. Since the concept sharing platform cannot access the original training images of the concept, it is essential to introduce a sampling process. The nature of iterative denoising in diffusion results in a lengthy backpropagation distance and significant memory and computation costs if we keep the gradient undetached. To make the watermark training viable, we maintain differentiability only during the last k steps of the sampling process (e.g., DDPM in Eq. (12) and Eq. (13)). Previous studies on diffusion models inspire this approach (Balaji et al., 2022): the model mainly focuses on the image layout in the early stage of diffusion model sampling, and gradually shifts the attention to producing high-fidelity visual outputs on details in the later stage.

When the gradient is calculated, it is influenced only by the last k steps. However, the output after the update, $E_{\theta+\triangle\theta}$, affects all sampling steps. Our solution is to keep a small learning rate and the sampling process online, to ensure that this misalignment can be addressed in the subsequent sampling process. During the training process, we let the diffusion model simultaneously forward multiple embeddings in a single mini-batch, including an original embedding and several embeddings with different randomly generated bit strings as watermark messages. The generated images from the watermarked embeddings must reflect the same concept as the original embedding. A compelling model is Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021). After being pre-trained on 400 million image-text pairs, it is highly effective at reflecting the similarity between high-dimensional semantics of images. We employ the CLIP model to calculate the cosine-similarity loss, which constrains the similarity between the watermarked image and the original image:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{N} \sum_{i=1}^{N} \frac{\langle E_{\text{CLIP}}(I^c), E_{\text{CLIP}}(I_i^w) \rangle}{\|E_{\text{CLIP}}(I^c)\|_2 \cdot \|E_{\text{CLIP}}(I_i^w)\|_2}, \tag{5}$$

where I^c and I^w denote the clean image and watermarked image, E_{CLIP} denotes the CLIP vision encoder.

4.2.3 Watermark Extraction

In the watermark extraction phase, the decoder D receives watermarked images generated by the diffusion model and aims to extract the embedded watermark from these images. We employ

EfficientNet-B3 (Tan & Le, 2019) pretrained on ImageNet (Deng et al., 2009) as our watermark decoder. We define q as the number of the message-bits encoded into the watermark and turn the output dimension of D into size $2q$. Then, we compute the Binary Cross Entropy (BCE) between the output logits l_j of the decoder D and the pre-defined watermark message $\mathbf{m}$:

$$l_j = \mathrm{softmax}(D(I^w)^{(2j-1)}, D(I^w)^{(2j)}), \tag{6}$$

$$\mathcal{L}_{\mathrm{Msg}} = \frac{1}{q}\sum_{j=1}^{q} \mathrm{BCE}(\mathbf{m}^{(j)}, l_j). \tag{7}$$

During the inference stage, we only need to compare the values of l_{2j-1} and l_{2j} to determine whether the bit $\mathbf{m}^{(j)}$ at index j should be 0 or 1. In addition, we assign an all-zero message for images generated by the original concept.

4.2.4 Training Details

In the training phase, we design a progressive loss function to tackle the challenges in concept watermarking. In the early iterations of training, we add a strong $\mathcal{L}_{\mathrm{Reg}}$ term to ensure that the output of the encoder quickly fits the Textual Inversion embedding, which contains the original concept. After that, we remove the $\mathcal{L}_{\mathrm{Reg}}$ term and only keep $\mathcal{L}_{\mathrm{CLIP}}$, $\mathcal{L}_{\mathrm{Msg}}$ and $\mathcal{L}_{\mathrm{Cst}}$. Specifically, we maintain a counter u that is decreased by 1 when $\mathcal{L}_{\mathrm{Reg}}$ is below a specified threshold h. When the counter becomes less than 0, it indicates that the encoder's output is at a favorable starting point. At this point, we remove the $\mathcal{L}_{\mathrm{Reg}}$ term. Experimentally, this training strategy significantly accelerates the fitting speed and produces better results. In general, our loss function can be formulated as follows:

$$\mathcal{L}_{\mathrm{Total}} = \begin{cases} \mathcal{L}_{\mathrm{CLIP}} + \mathcal{L}_{\mathrm{Msg}} + \lambda\mathcal{L}_{\mathrm{Reg}} + \mu\mathcal{L}_{\mathrm{Cst}} & \text{if } u > 0 \\ \mathcal{L}_{\mathrm{CLIP}} + \mathcal{L}_{\mathrm{Msg}} + \mu\mathcal{L}_{\mathrm{Cst}} & \text{else ,} \end{cases} \tag{8}$$

where λ, μ are hyper-parameters to weight each term.

4.3 Verification Stage

In the verification stage, the concept sharing platform leverages the decoder D to extract the watermark messages from the suspicious images I^x, and then traces back to the malicious users based on its record of watermark bits and user identification. Our watermarks exhibit strong generalization and robustness. Even the suspicious images are generated with different prompts, diffusion models, and configurations (including sampler, sampling steps, CFG scales), the decoder is still able to extract the watermarks from them accurately if they are from the watermarked embeddings.

Note that **our method handles each concept individually, using different watermark decoders for different concepts**. If a unified decoder were used, considering a scenario where two concepts ($\mathbf{m}_1$ and $\mathbf{m}_2$) are combined in the same image, attempting to extract either $\mathbf{m}_1$ or $\mathbf{m}_2$ would lead to an ill-defined problem. Conversely, if different decoders are used, then decoder D_1 can extract $\mathbf{m}_1$ and decoder D_2 can extract $\mathbf{m}_2$ from the same image.

5 Experiments

5.1 Experiment Settings

Dataset. We collected and curated various types of concepts from online sources as our test subjects. Among them, 3 concepts are specific individuals, 2 are related to art styles, and 2 are about certain objects, including vehicles and rare items. The types of the tested concepts cover mainstream Textual Inversion concept types to a large extent.

For the prompt to generate images, we apply basic prompts such as ("A photo of S*") as references and employ the GPT-3.5 model (Ouyang et al., 2022) to generate more prompt templates that encompassed a wider range of scenes. This approach can involve more prompts in the training process, allowing the watermark to be preserved better in the generated images when the concept is applied to different prompts. More details of prompt generation can be found in Appendix A.15.

Previous Methods. Currently, there are no watermarking methods dedicated to concept watermarking. People may first consider whether existing watermarking methods can be generalized to this new scenario. We added watermarks to the images in the training dataset for Textual Inversion, then trained the Textual Inversion model to learn the watermarks from the dataset. Upon completing the training, we sampled images and attempted to extract the watermarks from them. We conducted experiments using DWT-DCT-SVD (Rahman, 2013), RivaGAN (Zhang et al., 2019), StegaStamp (Tancik et al., 2020), and the latest methods CIN (Ma et al., 2022) and RoSteALS (Bui et al., 2023) as representative image watermarks.

Evaluation Metrics. To evaluate our proposed method, we adopt three types of metrics as follows: *Watermark Extraction Ability*, *Visual Fidelity*, and *Textual Editability*.

More details are provided in Appendix A.1.

5.2 Effectivness & Fidelity

5.2.1 Effectivness

Table 1 shows our method and the previous watermarking approaches in terms of the bit accuracy (Bit Acc.) and true positive rate (TPR) at the controlled false positive rate (FPR). It shows that previous watermarking methods could not perform well on this task.

For former methods, their primary failure is due to watermark patterns manifested at the pixel level and embedded in high frequency for imperceptibility. In contrast, Textual Inversion operates in a highly compressed semantic space. Therefore, these watermark patterns easily get lost during TI training. Our method directly adds watermarks to the existing textual embeddings, placing the watermark at the concept level, which makes it successful.

Table 1: Evidence shows that previous watermarking methods failed on this task. Our methods can effectively inject watermarks into a highly compressed embedding space.

Method	**Bit Acc.(%)↑**	**TPR(%)↑**	**CLIP-T↑**	**CLIP-I↑**
Original TI	-	-	25.97	81.70
DWT-DCT-SVD (Rahman, 2013)	49.88	0.0	24.80	81.61
RivaGAN (Zhang et al., 2019)	47.80	0.1	24.28	81.33
StegaStamp (Tancik et al., 2020)	47.31	0.0	21.80	78.38
CIN (Ma et al., 2022)	51.85	0.0	25.08	80.54
RoSteALS (Bui et al., 2023)	55.93	0.1	25.01	81.62
Ours	99.75	98.91	25.04	80.54

5.2.2 Visual Fidelity

Figure 3 illustrates qualitative examples of the images generated by watermarked Textual Inversion embeddings and the original Textual Inversion embeddings. Perceptually, they are indistinguishable at the concept level, which qualitatively illustrates our approach does well in adding watermarks to concepts while maintaining their reconstruction. More visual results are provided in Appendix A.16.

In Table 1, we present the quantitative results, where the calculation for image alignment (CLIP-I) of the original concept was computed by generating 128 images independently using the original concept with the basic prompt and subsequently calculating pair-wise clip similarity between the first 64 and the last 64 generated images. It sets an upper bound for this metric since the ultimate goal is to be as similar to the original concept as possible. Consistent with the perception of visual results, our method yields results that are very close to the upper bound.

5.2.3 Textual Editability

The right side of Figure 3 showcases the editability of the watermarked concept. Using different prompts, the watermarked concept can generate images that align well with the textual descriptions, just as the original concept does. Considering the numerical results, Table 1 shows that our approach achieves favorable text alignment (CLIP-T) scores, indicating that the impact of our watermarks on editability is minor. Although the watermark information could not be accurately extracted, the image watermarking methods also demonstrate good editability in the Table 1. The good results can be attributed to the fact that the added watermarks did not significantly impact the training process. As a result, the generated Textual Inversion embeddings closely resemble the original Textual Inversion embeddings, leading to very similar performance in the evaluation metrics.

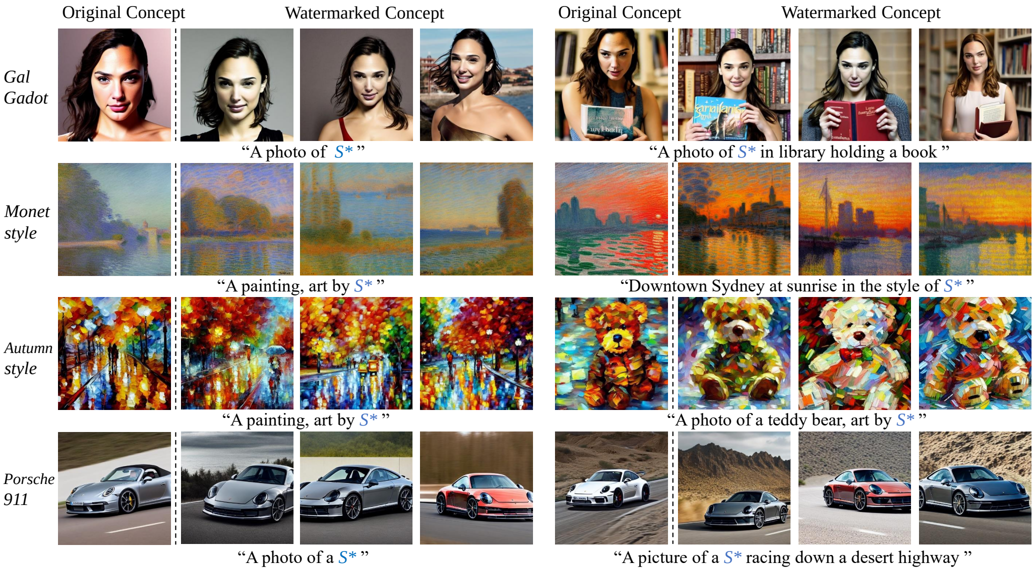

Figure 3: Comparison between images generated by the original concept and the watermarked concept under basic prompts and diverse editing prompts. We showcase three categories: person, style, and object. **Left:** The results obtained from basic prompts demonstrate excellent preservation ability of original concept semantics. **Right:** The results from diverse editing prompts showcase that the concepts with added watermarks maintain the same level of editability as the original concepts.

Table 2: Robustness against different diffusion configurations. For each setting, we showcase the average results. The gray cell denotes the default setting.

Configurations		Bit Acc.(%)↑	TPR(%)↑	CLIP-I↑
Default		99.75	99.89	80.54
Diverse Editing Prompts		98.51	97.51	-
Sampler	DDIM	99.75	99.89	80.54
	DDPM	99.36	99.41	80.21
	DPM-S	99.11	99.10	79.70
	Euler	99.75	99.74	80.15
Sampling Steps	14	98.55	99.10	80.05
	25	99.75	99.89	80.54
	38	99.33	100.0	79.52
	50	99.78	100.0	79.56
CFG Scales	5.0	99.11	99.10	80.48
	7.5	99.75	99.89	80.54
	10.0	99.56	100.0	79.89
SD Versions	SD v1.5	99.75	99.89	80.54
	SD v1.4	98.58	99.55	80.27
	Deliberate (XpucT)	93.43	87.39	81.07
	Chilloutmix (Anonymous)	91.19	79.68	79.54

5.3 Robustness

We comprehensively assess the robustness of our method against various diffusion sampling configurations and post-processing of generated images. In this section, we explore various settings for the diffusion sampling process. We consider that users can utilize different prompts, samplers, sampling steps, Classifier-Free Guidance (CFG) scales, and different SD versions. Table 2 shows strong robustness against different configurations. More details can be founded in Appendix A.3.

5.3.1 Post-processing on Generated Images

Images typically undergo numerous transformations and processing during transmission. We evaluated common post-processing methods: color jitter, crop & resize, rotation, blur, Gaussian noise, JPEG compress, and sharpness. Referring to Table 3, our method exhibits acceptable robustness

Table 3: Robustness against various post-processing on generated images.

Post-processing	Bit Acc.(%)↑	TPR(%)↑	CLIP-I↑
None	99.75	99.89	80.54
Color Jitter	98.55	98.42	77.75
Crop & Resize	99.28	97.93	79.65
Rotation	96.99	97.25	71.41
Blur	99.56	99.27	80.24
Gaussian Noise	95.43	97.15	79.04
JPEG Compress	98.16	97.89	80.08
Sharpness	96.88	98.15	77.34

in most cases. Some illustrations and detailed settings of post-processings can be found in Appendix A.11.

5.4 Ablation Studies

We conducted many ablation studies on our pipeline, conducted on one concept for demonstration.

The Influence of Encoder Architecture.

In addition to U-Net, we further explore other options, including MLP and ResNet, with more details of implementation in Appendix A.14.

Table 4: The influence of encoder architectures, watermark bits and gradient-preserved steps. The gray cell denotes the default setting.

Settings	Bit Acc.(%)↑	TPR(%)↑	CLIP-T↑	CLIP-I↑
Architectures				
U-Net	99.11	98.61	23.59	79.02
ResNet	95.63	98.04	23.46	80.10
MLP	73.90	7.03	22.71	81.16
Watermark Bits				
4	99.91	100.0	23.01	81.79
8	99.86	99.61	22.80	81.25
12	99.29	98.83	24.01	80.49
16	99.11	98.61	23.59	79.02
Gradient Steps				
1	68.50	18.41	24.85	81.19
2	83.19	23.14	24.08	80.15
3	99.11	98.61	23.59	79.02

From Table 4, we can observe that the MLP encoder under 16-bit settings achieves a relatively low extraction accuracy, making it unsuitable for practical use. Furthermore, it undergoes a significant decrease in CLIP-T, signifying a considerable loss in editability, which is unacceptable. While the ResNet encoder shows excellent fidelity, its extraction accuracy and CLIP-T are lower than U-Net.

We believe that U-Net's long-range skip connections assist in effectively constraining the watermarked Textual Inversion embedding within a region of high editability, which is why it outperforms ResNet.

The Influence of Watermark Bits We evaluated our pipeline's performance by embedding 4, 8, 12, and 16 bits into Textual Inversion embeddings, examining the extraction ability, fidelity, and editability. Table 4 presents our results. As the number of embedded bits increased, we observed a slight decrease in the extraction accuracy and fidelity. However, our method consistently maintains high performance within the tested range.

The Influence of Gradient-preserved Steps We examined the impact of retaining different numbers of gradient-preserved steps on the results. For Gradient backward steps=0 and 1, we did not wait for the loss curve to converge before stopping since its loss curve descended very slowly. We trained all settings for 10,000 steps. In Table 4, we observe that retaining more gradient backward steps leads to better results. Our current GPU limitations prevent us from increasing the number of gradient-preserved steps, but we anticipate further room for improvement in the future.

6 Conclusion

In this paper, we point out the demand for guarding concept sharing (i.e., Textual Inversion). To address it, we propose the novel *concept watermarking*, together with new training strategies and architecture design. We conduct extensive experiments to justify its practicability in concept-sharing scenarios, in terms of fidelity, effectiveness, traceability, and robustness. Moreover, adaptive attacks and ablation studies are also considered.

Aside from the technical aspects, we also advocate for the co-evolution of legislation and technology. By doing this, we hope to promote the coexistence of concept sharing and generated content safety, paving the way for a time when original ideas can flourish while still being safeguarded.

References

The complex world of style, copyright, and generative ai. https://creativecommons.org/2023/03/23/the-complex-world-of-style-copyright-and-generative-ai/.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*, 2023.

Anonymous. https://civitai.com/models/6424/chilloutmix.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2021.

Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 933–942, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Civitai. https://civitai.com/.

Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, and Jiliang Tang. Diffusionshield: A watermark for copyright protection against generative diffusion models. *arXiv preprint arXiv:2306.04642*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Stable Diffusion. https://github.com/CompVis/stable-diffusion.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435*, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2022.

Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *arXiv preprint arXiv:2302.12228*, 2023.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2672–2680, 2014. URL `https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html`.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Huggingface. https://huggingface.co/.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1931–1941, 2023.

Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Watermarking diffusion model, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1532–1542, 2022.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pp. 16784–16804. PMLR, 2022.

OpenAI. https://openai.com/blog/moving-ai-governance-forward.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Patreon. https://www.patreon.com/aitrepreneur/posts.

prompthero. https://prompthero.com/.

Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3403–3417, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Md Maklachur Rahman. A dwt, dct and svd based watermarking technique to protect the image piracy. *International Journal of Managing Public Sector Information and Communication Technologies*, 4(2):21, 2013.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Elad Richardson, Kfir Goldberg, Yuval Alaluf, and Daniel Cohen-Or. Conceptlab: Creative generation using diffusion prior constraints. *arXiv preprint arXiv:2308.02669*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2117–2126, 2020.

David Thiel. Identifying and eliminating csam in generative ml training data and models. Technical report, Technical report, Stanford University, Palo Alto, CA, 2023. URL https://purl ..., 2023.

Kent Walker. https://blog.google/outreach-initiatives/public-policy/our-commitment-to-advancing-bold-and-responsible-ai-together/.

Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. Hairclip: Design your hair by text and reference image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18072–18081, 2022.

Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Gang Hua, and Nenghai Yu. Hairclipv2: Unifying hair editing via proxy feature blending. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23589–23599, 2023.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Zejia Weng, Xitong Yang, Ang Li, Zuxuan Wu, and Yu-Gang Jiang. Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In *ICML*, 2023.

Whitehouse. https://www.whitehouse.gov/briefing-room/statements-releases/2023/07/21/fact-sheet-biden-harris-administration-secures-voluntary-commitments-from-leading-artificial-intelligence-companies-to-manage-the-risks-posed-by-ai/.

XpucT. https://civitai.com/models/4823.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. 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.

Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019.

Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. *arXiv preprint arXiv:2310.11868*, 2023.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

A Appendix

A.1 Experiment Setting

Evaluation Metrics. To evaluate our proposed method, we adopt three types of metrics as follows:

- **Watermark Extraction Ability.** We calculated the bit accuracy for N=1,000 images, where we randomly selected 4 different watermark bit strings, and for each string, we sampled 250 images. For the calculation of the True Positive Rate (TPR), we consider our method as a single-bit watermark, with a fixed watermark s. We extracted 10,000 clean images using the decoder, setting a threshold τ such that samples with a bit accuracy higher than this value are considered false positives. We adjusted τ to control the False Positive Rate (FPR) to be less than 1×10^{-3}. To mitigate the effects of randomness, we perform 4 trials with different s and compute the average TPR. The metrics can be formulated as follows:

$$\text{BitAcc}(s, s') = \frac{1}{|N|}\sum_{i=0}^{|N|} M(s_i, s'_i)/\text{len}(s_i) \tag{9}$$

$$\text{TPR}(\tau) = |N(M(s, s') > \tau)|/|N|, \tag{10}$$

  where $M(s, s')$ denotes the number of matching bits between s and s', N denotes the set of all test images, and $|\cdot|$ represents the number of elements in the set.
- **Visual Fidelity.** We aim for the images generated by the watermarked concept to have high visual similarity to those generated by the corresponding pristine concept. To assess this, we follow the methodology outlined in (Gal et al., 2022), employing image alignment (CLIP-I), which is determined by the cosine similarity of CLIP image embeddings between two images. Specifically, we calculate the CLIP-I between 64 watermarked images and 64 clean images, using basic prompts as the basis for generation.
- **Textual Editability.** Preserving the editability of textual inversion itself, which allows for modifying the generated content using prompts, is also crucial. To achieve this, we produce a set of images using prompts that depict a variety of scenes. These range from simple descriptions ("A photo of S*"), to style changes for non-style concepts ("A colorful graffiti of S*"), and to compositional prompts ("A photo of S* playing guitar in the forest"). We measure the alignment between the generated images and the given prompts, ensuring that the alignment does not decrease with the watermark add-on. For this, we use the text alignment (CLIP-T), which is used in custom diffusion(Kumari et al., 2023) and many other works. It is calculated by given prompts and corresponding images, computing text-image similarity in CLIP feature space.

Implementation Details. We conducted a survey and found that the number of tokens commonly used in Textual Inversion is much greater than 5 (see Appendix A.12). Therefore, we encode **16 bits** watermark information into 5 token-length embeddings by default. We set the gradient preserved steps k to 3, λ value to 10, and the μ value to 1×10^{-3}. The counter u was set to 200, and threshold h is set to 1×10^{-2}. We used the Adam(Kingma & Ba, 2015) optimizer with a batch size of 8. The initial learning rate was set to 1×10^{-4} and we use 2 gradient accumulation steps. We chose Stable Diffusion v1.5 as our base model due to its widespread usage and we adopt fp16 for it. All our experiments can be conducted on a single A6000 GPU. During the training process, we utilized DDIM (Song et al., 2020) as our sampling method. Although we used DDIM during training, the watermark exhibited robustness across various samplers after training completion. Further details can be found in the Sec. 5.3.

A.2 Preliminary: Diffusion Model

Recent advances in generative AI have facilitated the emergence of new powerful models, including Variational Autoencoders (VAE) (Kingma & Welling, 2013), Generative Adversarial Networks (GAN) (Goodfellow et al., 2014), Flow-based Models (Rezende & Mohamed, 2015), and Diffusion Models (DM) (Sohl-Dickstein et al., 2015). Among them, DMs have achieved state-of-the-art results in image synthesis. They employ non-equilibrium thermodynamics principles to gradually transform a simple prior distribution q_T into a complex one q_0 over a preset maximum number of timesteps T in the diffusion process.

The most prevalent variant of DMs is the Latent Diffusion Model (LDM) (Rombach et al., 2022), which runs the diffusion process in the latent space, making its training and inference more efficient. Specifically, LDM utilizes an image encoder $\mathcal{E}$ to convert an image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ into a latent representation $\mathbf{z}$, i.e., $\mathbf{z} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{h \times w \times c}$. Simultaneously, an image decoder $\mathcal{D}$ reconstructs the image from the latent representation $\mathbf{z}$ in a reverse fashion, i.e., $\mathbf{x} = \mathcal{D}(\mathbf{z})$. Additionally, a conditional denoising autoencoder $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})$ generates images from a given specific text $\mathbf{y}$ as a condition, where $\mathbf{z}_t$ signifies the latent representation at a particular time step $t \in \{1, ..., T\}$.

During training, a squared error loss $\mathcal{L}$ is used to compel LDM to denoise the latent representations $\mathbf{z}_t := \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$ as follows:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}_0, \boldsymbol{\epsilon}, t, \mathbf{y}} \left[\left\| \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t, t, \mathbf{y} \right) - \boldsymbol{\epsilon} \right\|_2^2 \right], \tag{11}$$

where α_t and σ_t represent the parameters of the diffusion process, $\boldsymbol{\epsilon}$ is sampled from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})$ is implemented as a U-Net (Ronneberger et al., 2015) conditioned on time-steps, and text vector. The text encoder, often utilizing the CLIP (Radford et al., 2021) text encoder, compresses the text prompt $\mathbf{y}$ into a vector, which is then fed into the U-Net. Given the trained DM $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})$, the sampling procedures can be represented as follows:

$$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{12}$$

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{z}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{y})) + \sigma_t \boldsymbol{\epsilon}, \tag{13}$$

where $\bar{\alpha}_t := \Pi_{i=1}^t \alpha_i$. Various sampling methods, including Denoising Diffusion Implicit Models (DDIM) (Song et al., 2020), Analytic-DPMS (Bao et al., 2021), Euler (Karras et al., 2022) and DPM-Solver (Lu et al., 2022), can utilize the trained model to sample more efficiently and achieve higher-quality results.

This paper, we focus on the most popular LDM, Stable Diffusion (SD) (Diffusion). SD models are open-sourced and available on the official website (Diffusion). As demonstrated in prior works (Qu et al., 2023; Zhang et al., 2023; Thiel, 2023), existing SD models have the inherent capability of generating harmful content given the sensitive prompts. By integrating with the personalization solution, they can further output illegal images with any personalized concept. This is the threat we aim to mitigate. As mentioned above, SD has diverse options for the sampling. This poses a great challenge for designing a robust concept watermarking method as the embedded watermark needs to remain effective across different diffusion sampling configurations.

A.3 Diffusion Sampling Configurations

Different Prompts. To test different types of concepts, we generate distinct prompts for each category. Broadly, our concepts fall into three categories: person, style, and object. Since these categories exhibit significant differences we aim to ensure prompt diversity. Thus for each category, we provide an instruction template and employ GPT-3.5 to generate specific prompts. We generated 64 prompts for each category and sampled 256 images for each concept under different watermark messages. To be specific, we randomly selected 4 different bit strings to get the corresponding watermarked embeddings, and sampled 64 images for each. More details can be found in Appendix A.15. As shown in Table 2 despite the high diversity of the prompts, our approach still manages to maintain a good extraction rate.

Different Samplers. In the denoising process, various types of samplers can be employed. In our study, we opted for DDIM(Song et al., 2020), DDPM(Ho et al., 2020), DPM-solver (DPM-S)(Lu et al., 2022), and Euler Scheduler(Karras et al., 2022). Among these, DPM-solver and Euler Scheduler are currently the most commonly used and widely adopted samplers. For each sampler, we generated 64 images for our evaluation. We found that changing the sampler has minimal impact on the watermark extraction (see Table 2) since our watermark existed in the text encoder which won't be affected by the sampling method.

Different Sampling Steps. Users have the flexibility to use different sampling steps for content generation. Thus, we evaluated watermark extraction performance under various sampling step settings. We observed that when the sampling steps are fewer than 14, the DDPM sampler fails to produce high-quality results. Consequently, we chose 16 distinct sampling step values that were almost uniformly spread between 14 and 50. For each step value, we generated 4 different images, resulting in a total of 64 images, computed the accuracy of watermark extraction, and CLIP-I for visual fidelity. Based on the results, as the sample steps increase from 14 to 25, the quality of the samples progressively improves, and the accuracy also increases. Beyond 25 steps, the improvement in sample quality becomes marginal and the extraction rate stabilizes (see Table 2).

Different CFG Scales. Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) is a sampling method that offers greater flexibility compared to classifier guidance and has been proven to achieve better generation quality. We selected 16 different CFG scale values evenly spaced between 5.0 and 10.0. Similarly, for each CFG scale value, we generated 4 different images, resulting in a total of 64 images, and computed the Bit accuracy and True Positive Rate of watermark extraction, CLIP-I for visual fidelity. We observed that although there is a certain tendency for the watermark extraction and CFG scales, our extraction ability remains at a relatively high level. Furthermore, excessively low CFG scales may lead to unusable results.

Different SD Versions. Users have the flexibility to choose different versions of Stable Diffusion, such as SD v1.4 or other fine-tuned versions of SD models. As shown in Table 2, our method still maintains good accuracy. For *Deliberate* (XpucT) and *Chilloutmix* (Anonymous), although the accuracy has decreased, it still maintains a high accuracy.

A.4 The Capacity of the Proposed Method

The empirical results demonstrate that our method achieves a remarkable 99.75% Bit Accuracy for the evaluated concepts using 16-bit settings. This makes it suitable for accommodating up to about 10,000 users. However, we recognize that 16 bits is relatively small for a typical watermarking scheme. The capacity of the watermark can be further increased by allocating a greater number of tokens in the Textual Inversion training at a relatively low additional cost. Moreover, expanding the number of gradient steps k and increasing the sizes of the encoder and decoder models can potentially improve scalability.

A.5 The Computational Cost

Training a pair of encoder-decoder requires approximately 2 GPU hours on an A6000 GPU. We did not use a universal decoder due to the following considerations: concept composition is common in the use of Stable Diffusion, and for the combination of secret messages $\mathbf{m}_1$ and $\mathbf{m}_2$, this would become an ill-defined problem. Neither $\mathbf{m}_1$ nor $\mathbf{m}_2$ could be the correct answer. Based on this, we believe training encoder-decoders for specific concepts is necessary.

A.6 Multiple Concepts Reactions

In Stable Diffusion, there's a possibility of using multiple concepts during the process of generating images. A pertinent research question arises: will there be any conflicts between the concepts added as watermarks? We tested different scenarios of the combined use of style-type concepts (e.g., *Monet* and *Autumn*) and person-type concepts (e.g., *Trump* and *Gal*), as shown in Table 5. We observed that when combining person with person it is normal for two individuals to appear in a picture, resulting in both having high extraction rates. However, when person and style are mixed, there's a decrease in accuracy for both but still feasible. When two style-type concepts appear in the prompt simultaneously, the accuracy of one becomes very low. This is because, for person mixed with style, since the concepts expressed by the two don't clash in the image, watermark extraction is still feasible. But for two styles, one style is often dominated by the other and doesn't get represented in the image.

Figure 4 presents the failure cases in multi-concept scenarios. Sometimes certain concepts tend to be suppressed by others, rendering them invisible in the image. Under these circumstances, the extraction rate of the dominated concept is typically lower. This validates that our watermarks reside in a high-dimensional concept space. We believe that even if a malicious user employs the concept with our watermark, it is reasonable not to extract it if the corresponding concept doesn't exist in the generated image.

In summary, when a concept can be fully expressed in an image, even if the image contains other concepts, our concept watermarking remains effective.

A.7 Comparison for Different Sources of TI

The process of training Textual Inversion requires extensive data collection and experience in parameter changing. Default settings and poor training data can lead to suboptimal results. In Figure 5, we display the generated results of Textual Inversion trained under default settings, as well as the results of downloaded Textual Inversion.

A.8 Theoretical vs. Experimental False Positive Rates

We consider all the watermark approach as a single-bit watermark, with a fixed watermark s. A predetermined threshold value, τ, which ranges from 0 to k, is used to evaluate the watermark's presence. If the matching bits number $M(s, s')$, between the watermark in the image and the fixed watermark s meets or exceeds this threshold τ, it is concluded that the watermark is indeed present in the image.

In previous research Yu et al. (2021), it is commonly assumed that the bits extracted from the original image are independent and identically distributed (i.i.d) Bernoulli random variables with a parameter of 0.5. Based on this assumption, we can define the theoretical upper limit of the False Positive Rate (FPR). This probability can be expressed using the regularized incomplete beta function $B_x(a; b)$,

$$FPR(\tau) = P(M(s, s') > \tau) = \sum_{i=\tau+1}^{k} \binom{k}{i} \frac{1}{2^k} = B_{\frac{1}{2}}(\tau + 1, k - \tau). \tag{14}$$

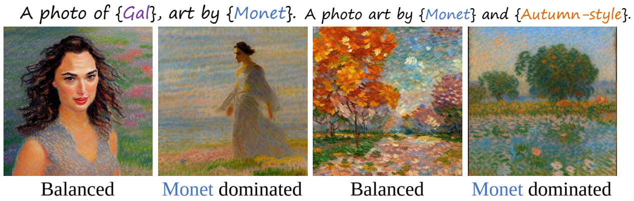
A photo of {Gal}, art by {Monet}.
A photo art by {Monet} and {Autumn-style}.
Balanced
Monet dominated
Balanced
Monet dominated

Figure 4: The image displays the failure cases in multi-concept scenarios. Moving from left to right, the second image shows the dominance of the style concept over the person concept. The third image displays a unique balance between two style concepts, resulting in a distinct style, while the fourth highlights the dominance of one style over the other.

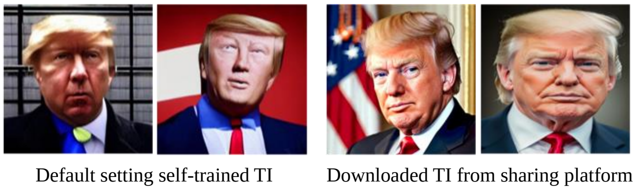
Default setting self-trained TI
Downloaded TI from sharing platform

Figure 5: Comparison of the TI trained using default settings and Downloaded TI which may incorporate more expertise.

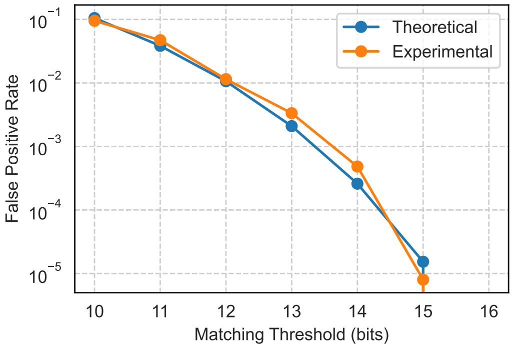

Figure 6: Comparison of the theoretical and experimental false positive rate using different watermark matching thresholds.

By setting the desired level for FPR, we can use the Equation above to determine the minimum threshold value τ required for watermark matching.

However, the i.i.d assumption might not be valid for our trained watermark extractor, suggesting that the watermark bits extracted could be dependent. Consequently, we experiment to estimate the detection False Positive Rate. We select 100,000 vanilla images from the MSCOCO dataset training set, resize them to 512×512 pixels, and then extract a watermark from each image, which we denote as m_{clean}. Then we randomly generated 10 distinct ground truth watermark messages, denoted as m_{gt}. The watermark matching threshold, τ, is chosen from 10 to 16. For every combination of the watermark message and watermark matching threshold, we match the extracted watermark messages with their respective ground truth watermark message and compute the FPR. A false positive is identified when a vanilla image's extracted message coincides with the ground truth message, such that $M(m_{clean}, m_{gt}) \geq \tau$.

Figure 6 in the paper visualizes and contrasts the FPR for both the estimated and the empirical calculations. It can be observed that the experimental results are very close to the theoretical values.

Table 5: Accuracy of extraction after the combination of two concepts.

Mixed Concepts	**Decoder**	**Bit Acc.(%)**	**TPR(%)**
Monet & Autumn	style (Monet) style (Autumn)	94.05 64.02	85.9 6.30
Trump & Autumn	person (Trump) style (Autumn)	94.95 87.39	89.4 70.2
Gal & Monet	person (Gal) style (Monet)	88.45 90.16	65.1 83.0
Trump & Gal	person (Trump) person (Gal)	95.28 96.81	95.4 96.5

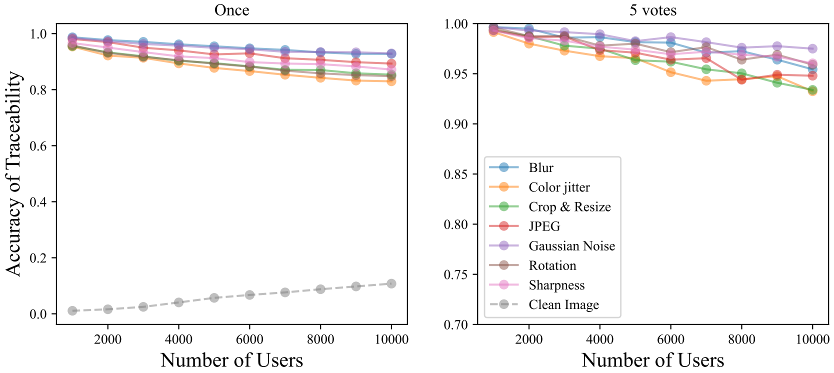

Figure 7: Traceability results for *concept watermarking* under the different number of users. **Left:** traceability accuracy for individual watermarked images and FPR for clean images. **Right:** traceability accuracy of 5 images voting for a single user.

In the main body, we examined the metric TPR@FPR=1×10^{-3}, which corresponds to a matching threshold of $\tau = 14$.

A.9 Traceability

The Traceability experiment aims to simulate N users downloading the watermarked concept, with N' users sharing the generated images online (N'=1,000 in our experiments), and each sharing 10 images, totaling 10,000 images. We then extract the watermark from these images. For each image, we calculate the bit accuracy between this image and the watermarks assigned to N users, $Acc = \{acc_1, acc_2, \ldots, acc_N\}$. If $max(Acc) < 1$, we consider it not to contain a watermark; otherwise, we consider the image to contain the watermark corresponding to $max(Acc)$. This allows us to calculate the Accuracy of Traceability by comparing the trace results with the original labels.

Figure 7 shows the traceability accuracy under different post-processing conditions. We examined two scenarios: treating each image independently, and considering images uploaded by the same user, performing bit-by-bit voting on the extraction results. For the former condition, even with N=10,000 users and no distortions, the accuracy is 94.82%. Although traceability is poorer under distortions, for the latter, if a user uploads 5 images and we perform bit-by-bit voting on the extraction results, the traceability accuracy can be largely increased to nearly 95%.

To measure whether clean images are correctly detected as unwatermarked, we provide the false positive rate (FPR) results for clean images using our method. We used a total of 10,000 clean images for detection. Same as the settings we mentioned above, we assume that there are N users assigned different watermarks. If the information extracted from clean images is recognized as one of these N watermarks, it is considered a false positive. The experimental false positive rate is presented as the gray dash line in Figure 7. For up to 5,000 users, we can control the FPR under 5%.

A.10 More Ablations

The Effectiveness of Contrastive Loss We tested the impact of Contrastive Loss on enhancing robustness against embedding distortion for one of the concepts. As illustrated in Table 6, incorporating this loss significantly enhances performance against embedding distortion. This improvement is due to the Contrastive Loss causing embeddings with different watermarks to be farther apart in direction, thus providing better protection against direct distortions applied to the embeddings.

A.11 Details of Post-processing Distortion

Figure 8 demonstrates the transformations we evaluated in the main body.

We utilized the `kornia` python library for the following transformations:

Table 6: The impact of the contrastive loss on the robustness against embedding distortion.

Embedding Distortion	**Contrastive Loss (Bit Acc.(%)↑)**	
	w/o	**w/**
None	99.22	99.11
Gaussian Noise	98.15	98.59
Rescale	92.28	95.60
Smoothing	93.61	95.99

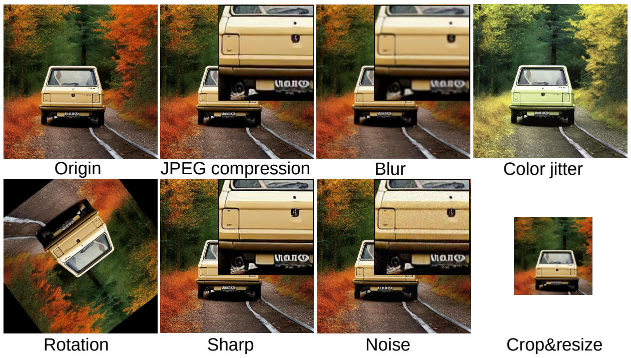

Figure 8: Transformations evaluated in post-processing robustness.

`Color jitter.` We modified the brightness factor, contrast factor, saturation to 0.3, and hue factor to 0.1.

`crop and resize.` We randomly extracted 384×384 blocks from the 512×512 images and resized them to 256×256.

`Gaussian blur.` We set kernel size of (3, 3) and sigma of (2.0, 2.0) on the images.

`Gaussian noise.` We added with a standard deviation of 0.05.

`JPEG compression.` We set with a quality setting of 50.

`Rotation.` We randomly applied to the images within a range of 0 to 180 degrees.

`sharpness.` We set the intensity to 10.

A.12 Token Numbers in Textual Inversions

We downloaded 100 Textual Inversions from Civitai and tallied their token counts (see Figure 9). The average token count for these 100 Textual Inversions is 8.87.

A.13 Details of Embedding Distortion

The attacker can also distort the download concept before generating images. We consider the following three distortion operations:

Adding Gaussian Noise. We added Gaussian noise to the embeddings with an intensity of $\sigma = 1 \times 10^{-1}$ which is relatively high as we examined 36 Textual Inversion embeddings downloaded from the internet, with the norm typically around 1×10^{-1}.

Rescaling Concept Embedding. We rescale the concept embedding by simply multiplying the embedding with a factor. In our experiments, we set this value to 0.4.

Smoothing Concept Embedding. We smooth the embeddings through a conv layer with the 1D kernel [0.2, 0.6, 0.2].

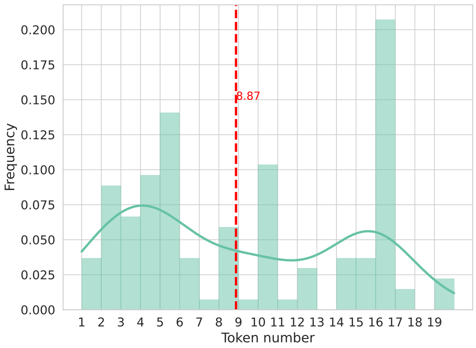

Figure 9: Statistical results of token numbers in 100 Textual Inversions.

A.14 Implementation Details of Different Encoder Architectures

MLP. For the implementation of the MLP encoder, we flatten the concept and then repeat the Message m to match the size of the embedding. The concatenated data is then sent to the MLP with a hidden size of 1536. The default MLP consists of 3 layers of Linear transformations and 2 layers of ReLU.

ResNet. For the implementation of the ResNet encoder, For the implementation of the ResNet encoder, we adopted the same strategy as with the U-Net, using a linear layer to map the bit message $\mathbf{m}$ to the same size as the embedding size d_{τ}. In contrast to U-Net, we utilized ResNet blocks here, retaining only the in-block skip connections. Considering the number of parameters, we did not use any existing ResNet models but defined a 10-layer ResNet, achieving a parameter size similar to that of the U-Net encoder.

A.15 Prompt Generation

Due to the existence of various types of concepts, a universal prompt is not applicable. Therefore, we created different prompt datasets for different types of concepts. We generate our training and testing prompt using GPT-3.5 by telling it with the following instructions:

[name] represents a (coarse class), take the following prompt as a reference, and generate (number) more prompts with various scenes and descriptions: (prompt example)

Here, (coarse class) should be replaced according to the category of the current concept, for example, person, style, car, crystalskull, etc. And (number) should be replaced by the number of prompts you want to generate. For concepts related to particular persons or objects, we chose (A photo of [name]) as our prompt example. For the style-like concepts, we decided (A painting, art by [name]) as our prompt example. For cars and other objects in paper, we use (A photo of a [name])

A.16 Additional Visual Results

See Figure 10.

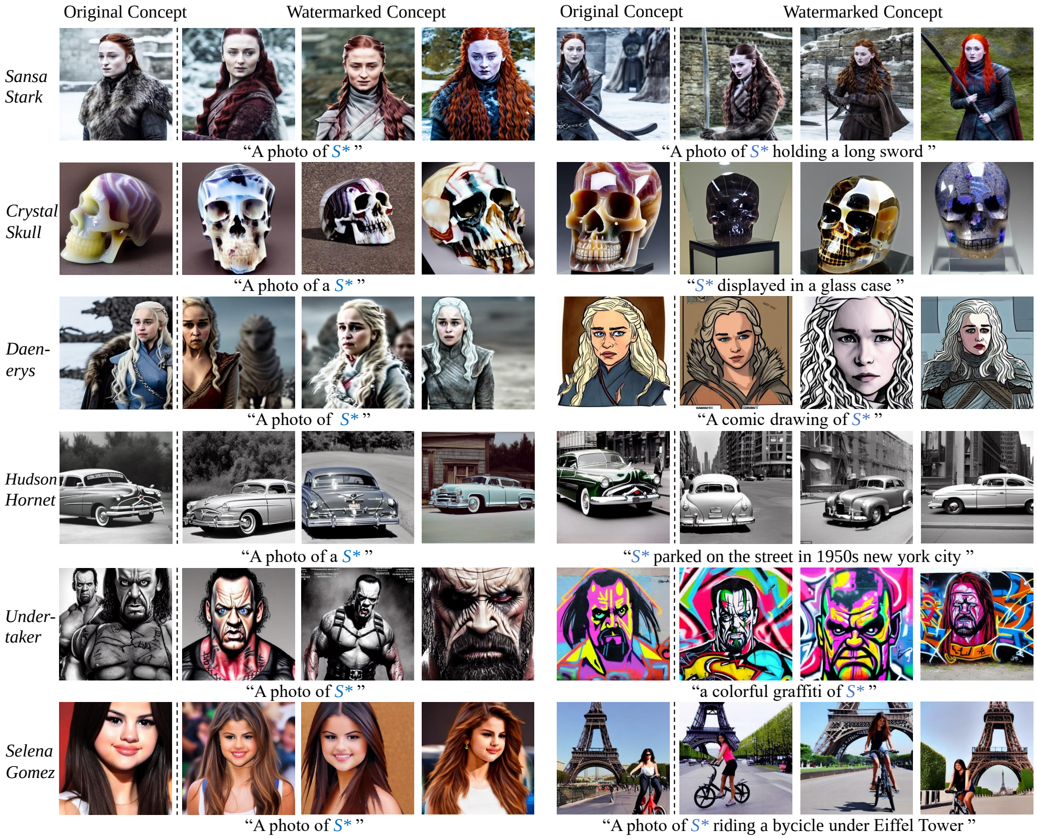

Figure 10: Additional visual results.