Scalable and Robust Watermarking for Diffusion Models via Task-Decoupled Mixture-of-Watermarks

Xiaorui Dai, Wei Wang, *Member, IEEE*, and Bo Wang, *Member, IEEE*

***Abstract*—With the growing deployment of text-to-image diffusion models in real-world applications, concerns about model copyright have become increasingly prominent. Model watermarking has emerged as a promising solution. However, existing methods often suffer from degradation of model fidelity and poor scalability in model distribution scenarios. Moreover, recent studies have shown that some watermarking approaches lack robustness against ambiguity attacks. To address these challenges, we propose TD-MoW, a lightweight and plug-and-play watermarking framework that enables robust and scalable watermark embedding. TD-MoW consists of a two-stage process: In the first stage, we independently optimize the word embeddings of the watermark trigger along with a low-rank adaptation (LoRA) module, effectively decoupling watermark learning from the original generation objective without modifying the base model. In the second stage, inspired by the mixture-of-experts (MoE), we introduce the MoW mechanism, which integrates multiple specialized watermark experts into the model. Through encoding optimization for watermarking, we enable n watermark experts to support up to 2^n users, reducing the complexity $\mathcal{O}(2^n)$ to $\mathcal{O}(n)$. Moreover, by leveraging gated activation, the watermark experts remain inactive during standard inference, thereby preserving generation fidelity. Extensive experiments demonstrate that our method not only achieves superior efficiency and generation fidelity compared to prior approaches, but also exhibits strong robustness.**

***Index Terms*—Diffusion models, mixture-of-experts (MoE), model copyright protection, model traceability, model watermarking.**

I. Introduction

TEXT-TO-IMAGE generative models [1], [2], [3], [4] have garnered significant attention in both academia and industry due to their ability to generate high-quality images [5]. Among them, latent diffusion models (LDMs) [3], [6], [7] have addressed the limitations of traditional generative adversarial networks (GANs) [8], [9] and variational autoencoders (VAEs) [10], particularly in terms of training stability and output diversity, by operating in a lower dimensional latent space instead of directly in high-dimensional image space. These advantages have led to their widespread adoption in fields such as creative industries, advertising, and automated content generation [11].

However, as LDMs become increasingly integrated into commercial services, concerns regarding intellectual property (IP) of the model itself have grown significantly [12], [13], [14]. Given the substantial computational resources and expertise required for their development, these models hold considerable value for researchers and developers [15]. Once the trained model is publicly released or shared within a restricted group, it becomes difficult to prevent unauthorized redistribution. Furthermore, leaked models may be used to generate content that violates ethical guidelines, infringes on copyrighted materials, or enables malicious applications, further complicating legal and regulatory efforts.

Watermarking technology [16], [17], [18], [19] offers a promising solution for the IP protection of diffusion models. Existing watermarking research on diffusion models can be categorized into two paradigms: watermarking of generated images and watermarking of diffusion models. Generated images watermarking aims to embed multibit watermark information into the generated image, thereby enabling both model and image ownership verification and traceability. Methods [20], [21], [22], and [23] investigate watermark embedding based on the initial noise, where the watermark is injected into the initial noise vector of the diffusion model, and DDIM inversion is used to reconstruct the embedded noise from the watermarked image. Other approaches incorporate watermarking by fine-tuning the different modules of diffusion model, such as fine-tuning the decoder of VAE [24], [25], [26], [27], [28] or the conditional denoising model U-Net [29], [30] to embed watermarks. However, these methods often rely on the original training data or diffusion model for watermarks extraction. Moreover, when the model is fine-tuned or image is compressed by attackers, the embedded watermark can be easily removed.

In contrast, diffusion model watermarking provides a more distinctive and verifiable identifier by embedding watermarks into the model itself through trigger-based mechanisms. These methods construct a set of trigger prompts paired with corresponding watermark images and fine-tune the model using this trigger set. Model ownership can then be verified by comparing the model's outputs on trigger prompts with refer-

Received 8 June 2025; revised 7 October 2025; accepted 8 January 2026. Date of publication 20 January 2026; date of current version 26 March 2026. This work was supported in part by the Application Fundamental Research Project of Liaoning Province under Grant 2025JH2/101330123, in part by the Open Research Fund of the National Key Laboratory of Cyber Space Behavior Cognition Technology under Grant CBCT2024KF02, and in part by the National Natural Science Foundation of China under Grant 62372452. *(Corresponding author: Bo Wang.)*

Xiaorui Dai and Bo Wang are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116081, China (e-mail: xiaoruidai@mail.dlut.edu.cn; bowang@dlut.edu.cn).

Wei Wang is with the New Laboratory of Pattern Recognition (NLPR), State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China (e-mail: wwang@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/JIOT.2026.3656382

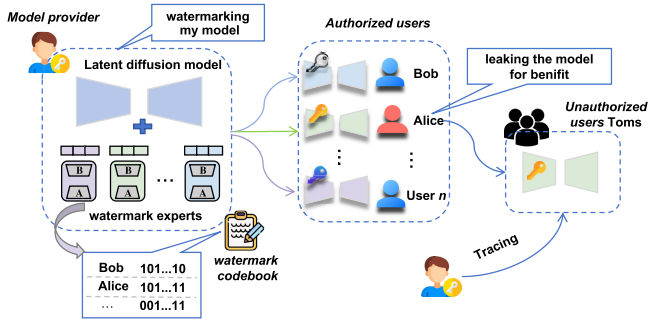

Fig. 1. Application scenarios of our watermarking framework.

ence watermark images. Methods [31], [32], and [33] have investigated text-to-image diffusion model watermarking, where rare tokens (e.g., "[V]") are paired with custom-designed images to form the watermark trigger set. The diffusion model is then fine-tuned using this set by adjusting the text encoder and U-Net, ensuring that only models containing the watermark generate specific images when queried with the trigger prompts.

Although trigger-based watermarking provides more distinctive watermark images, it still suffers from the following three limitations.

1) *Degradation of Model Fidelity:* These approaches fine-tune the key components of the LDM, such as the text encoder or the conditional denoising model, leading to a strong coupling between the watermark embedding task and the model's original generation task. This interference can adversely affect the quality of the generation.
2) *Vulnerability to Ambiguity Attacks:* The attacker can reverse-engineer the model to craft a forged prompt that differs from the original watermark prompt but still generates the same watermarked image. This ambiguity undermines the uniqueness of ownership attribution. Prior work [34] has demonstrated that some of the existing trigger-based watermarking approaches are inherently vulnerable to ambiguity attacks.
3) *Scalability Limitations in Model Distribution Scenarios:* Both the existing trigger-based and generated images watermarking are impractical in model distribution scenarios (Fig. 1) due to linearly scaling resource demands. Since watermark embedding must be performed individually for each distribution model, both the computational and local storage costs increase linearly with the number of users. For example, given the conditional denoising model of the stable diffusion 1.5 (SD1.5) with about 860 million parameters, and the SDXL [7] with more than 2.2 billion parameters, the time complexity and local storage required for watermark embedding scale as $\mathcal{O}(N)$ for N users. This linear growth poses significant challenges for large-scale model distribution scenarios.

To address these challenges, we introduce task-decoupled mixture-of-watermarks (TD-MoW), a scalable and robust watermarking framework designed to efficiently embed watermarks into text-to-image diffusion models in model distribution scenarios. Our framework consists of two stages: decoupling training and mixture-of-watermarks. In the first stage, we adopt a trigger-based watermarking approach by designing easily constructible watermark triggers based on combinations of uncommon tokens, which are added into the vocabulary as independent tokens. This design enables the theoretical generation of a large number of distinct triggers. We then fine-tune both the word embedding of the watermark trigger and a low-rank adaptation (LoRA) module [35]. These components collectively form a watermark expert, enabling watermark embedding without altering the core architecture of the LDM, thereby effectively decoupling the watermarking task from the original generation objective. In the second stage, to further achieve scalable watermark in model distribution scenarios, inspired by mixture-of-experts (MoE) [36], we introduce the MoW mechanism. We train k watermark experts based on the first stage and encode them into $2^k - 1$ unique watermark identifiers, assigning each user a unique watermark identifiers. This ensures that the watermark's scalability grows exponentially with the number of available experts. We then use a predefined watermark encoding table, we integrate the corresponding watermark experts into the base model. During watermark verification, a lightweight gating network ensures that only the designated watermark expert is activated, preserving model fidelity while generating the target watermark image. To further improve stealthiness and robustness, we blend the trigger into natural language text and combine it with the gated activation mechanism, which significantly enhances the robustness of our approach against ambiguity attacks.

In summary, the main contributions of the proposed *TD-MoW* framework are as follows.

1) We propose a scalable and robust watermarking framework, TD-MoW, which performs watermark encoding optimization. Specifically, it enables only n lightweight watermark experts to support up to 2^n users, reducing the computational and local storage complexity from $\mathcal{O}(2^n)$ to $\mathcal{O}(n)$.
2) We design stealthy watermark prompts combined with gated activation, ensuring that watermark experts are only activated by specific inputs. This preserves the model fidelity and improves robustness against ambiguity attacks.
3) Extensive experiments demonstrate the effectiveness and practicality of TD-MoW for diffusion model ownership verification and leak traceability. Our method outperforms existing approaches in both efficiency and robustness against ambiguity attacks.

II. Related Works

A. Watermarking of Generated Images

Generated image watermarking primarily focuses on embedding imperceptible identifiers into all generated images, thereby achieving traceability of both the generated content and intellectual property of the model.

Several studies have explored *initial noise-based* watermarking. Wen et al. [22] first proposed the initial noise-based watermarking method of diffusion models, Tree-Ring, which embeds watermark into the frequency domain of the initial noise, using a circular embedding region to enhance robustness against rotation attacks. Ci et al. [21] extended Tree-Ring into a multikey watermarking framework by introducing a

discretization mechanism and a lossless embedding strategy, thereby improving robustness and enabling resistance to rotation attacks. Arabi et al. [20] introduced a two-stage detection framework capable of resisting both forgery and removal attacks. Yang et al. [23] proposed a lossless watermarking method that first encrypts the watermark information into a uniformly distributed representation using a cryptographic algorithm, and then transforms it into a Gaussian distribution via inverse transform sampling, thereby minimizing its impact on the generated images. However, relying on the approximate reversibility of DDIM requires the same diffusion model for both embedding and extraction. In practice, using the diffusion model itself poses challenges in terms of storage and deployment efficiency.

Other studies have explored the *fine-tuning-based* watermarking. Fernandez et al. [27] jointly fine-tuned a pretrained watermark decoder together with the image decoder of a diffusion model, enabling the generation of images embedded with a specific binary watermark sequence for model ownership verification and user traceability. Xiong et al. [28] designed an information encoder that transforms watermark data into an information matrix, which is then embedded into the intermediate representations of the image decoder to produce watermarked outputs. Kim et al. [26] introduced a mapping network that projects the watermark into a latent fingerprint image aligned with the diffusion model's internal dimensionality. Min et al. [30] employed a pretrained linear layer to transform watermark into intermediate features encoding the watermark. These features are then injected into the final sampling steps of the image generation process using a fine-tuned diffusion model, resulting in watermarked outputs. Feng et al. [29] trained a set of watermark-specific encoder and decoder modules in the latent space, and introduced an LoRA module to embed watermark information into the generated images effectively.

Most of the fine-tuning-based methods preserve the layout of the generated image. However, these methods often rely on the original training data and when the model is fine-tuned or the output image is compressed by adversaries, the embedded watermark can be easily removed. In contrast, diffusion model watermarking provides a more distinctive and verifiable watermark through trigger-based mechanisms.

B. Watermarking of Diffusion Models

Diffusion models watermarking typically embed watermark through a *trigger-based* mechanism, where the trigger set consists of input–output pairs. The watermark extraction and verification can be performed by querying the model without requiring access to its internal structure. Some studies have explored trigger-based watermarking in discriminative models [37], [38], [39], but research in diffusion models remains relatively scarce.

Peng et al. [40] proposed a watermarking method for unconditional diffusion models, WDP, which embeds watermarks by fine-tuning the model jointly on both the watermark image and the original training data. However, this embedding process incurs substantial computational overhead and leads to a noticeable degradation in model performance. Liu et al. [31] investigated watermark injection for text-to-image diffusion models by embedding trigger prompts at fixed positions within the input text to enhance watermark stealthiness. The model fine-tunes the original data and trigger sets. However, increasing the length of the trigger prompt leads to a significant drop in model performance. Zhao et al. [33] proposed a watermarking method that uses rare tokens (e.g., "[V]") combined with customized images to construct a trigger set. The diffusion model is then fine-tuned on this set by updating the text encoder and the U-Net, while incorporating the original model parameters as a regularization term to preserve the model's original performance. Compared to [31], this method enables more efficient watermark embedding. In contrast, Yuan et al. [32] did not incorporate the original model as a constraint during optimization. Instead, they added customized trigger prompts to the original vocabulary and separately fine-tuned the text encoder and the U-Net to achieve watermark embedding.

However, trigger-based studies make a tight coupling between the watermark embedding task and the original model leading to a degradation in generation performance. And some methods have been shown to be vulnerable to ambiguity attacks. Moreover, in model distribution scenarios, both the watermarking of generated images and trigger-based methods lack scalability, incurring significant computational and storage overhead. This work aims to address these critical challenges.

C. Ambiguity Attacks for Diffusion Model Watermarking

Ambiguity attacks have been extensively studied in the context of traditional digital watermarking [41], [42]. Unlike removal attacks that aim to erase embedded watermarks, ambiguity attacks seek to undermine ownership verification by injecting a forged watermark into the digital content. This causes the original watermark, embedded by the legitimate owner, to lose its uniqueness, thereby obscuring the rightful copyright attribution.

Diffusion model watermarking is similarly vulnerable to ambiguity attacks. Specifically, once the model is accessible, an adversary can reverse-engineer a new forged trigger prompt—distinct from the original one—that still produces the same watermark image. In such cases, both the legitimate owner and the adversary can present equally convincing evidence of ownership, rendering the watermarking mechanism ineffective for copyright protection. Yuan et al. [34] recently proposed an ambiguity attack to diffusion models. By leveraging discrete projection and adversarial optimization, they successfully attacked the watermarking method [33], generating forged trigger prompts capable of reproducing the original watermark image and thereby invalidating ownership claims. To defend against such attacks, they introduced a method [43] that embeds multiple trigger-based watermarks and protects them using key-based encryption. However, their experiments showed that embedding multiple watermarks leads to a significant degradation in model performance.

In this work, we propose a novel watermarking approach that effectively resists ambiguity attacks while preserving the generative quality of diffusion models.

III. Preliminary

A. Diffusion Models

1) Diffusion and Reverse Diffusion: The diffusion process gradually corrupts images by adding Gaussian noise over T steps, transforming a clean image x_0 into pure noise x_T. The forward diffusion is defined as follows:

$$q(x_t|x_0) := \mathcal{N}\left(x_t;\ \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{1}$$

where β_t is a noise schedule, $a_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$.

The reverse process is a Markov chain defined by Gaussian transitions

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t)\right) \tag{2}$$

where $\mu_\theta(x_t,t)$ is derived from the noise prediction ϵ_θ, and Σ_θ is either fixed or learned. This formulation enables iterative denoising from noise x_T to a clean sample x_0. A conditional denoising model ϵ_θ predicts the noise component at each step, the training objective minimizes

$$\mathcal{L} = \mathbb{E}_{t,x,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t,t)\|^2\right]. \tag{3}$$

2) Latent Diffusion Models: LDM is an efficient diffusion model that leverages latent space encoding to reduce computational complexity while preserving generative performance. LDM consists of three key components: a text encoder, an autoencoder, and a conditional denoising model U-Net.

The text encoder transforms the input text y into a high-dimensional semantic vector $\tau(y)$, which serves as conditional information to guide image generation. Taking the CLIP text encoder [44] as an example, the input text y is tokenized into a sequence of tokens $\{t_1, t_2, \ldots, t_n\}$, where each t_i is an integer index in the vocabulary V. These tokens are mapped to word embeddings via an embedding matrix M, and M is trainable

$$v_i = M[t_i], \quad M \in \mathbb{R}^{d\times m} \tag{4}$$

$$V^* = \{v_1, v_2, \ldots, v_n\}, \quad v_i \in \mathbb{R}^d \tag{5}$$

where v_i is the word embedding of token t_i, m represents the vocabulary size, and V^* denotes the word embedding set of input text y. These word embeddings are then fed into a multilayer text transformer encoder, producing a dense representation $\tau(y)$ that captures the semantic meaning of the text

$$\tau(y) = \text{Transformer}(V^*). \tag{6}$$

We only fine-tuning the word embedding v^* of the trigger prompt in our method.

The autoencoder maps high-dimensional images into a lower dimensional latent space and reconstructs them through a decoder

$$z_0 = E(x), \quad x \in \mathbb{R}^{H\times W\times C} \tag{7}$$

$$\hat{x} = D(z_0) \tag{8}$$

where E is the image encoder, D is the decoder, and z_0 denotes the latent variable.

The conditional denoising model progressively denoises in the latent space z, following a forward diffusion (noise addition) and reverse diffusion (denoising) process. The conditional information $\tau(y)$ is incorporated into the denoising process via a cross-attention mechanism. The diffusion process then occurs in this lower dimensional space

$$q(z_t|z_{t-1}) := \mathcal{N}\left(z_t;\ \sqrt{\bar{\alpha}_t}z_{t-1}, (1-\bar{\alpha}_t)\mathbf{I}\right). \tag{9}$$

The reverse process generates latent variable z_{t-1} from z_t guided by task-specific text, and the loss is then given by

$$p_\theta(z_{t-1} \mid z_t, \tau(y)) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t,t,\tau(y)), \Sigma_t) \tag{10}$$

$$L_{\text{DM}} = \mathbb{E}_{t,z_0,y,\epsilon\sim\mathcal{N}(0,1)}\left[\|\epsilon - \epsilon_\theta(z_t,t,\tau(y))\|^2\right] \tag{11}$$

where $z_t = (\alpha_t)^{1/2}z_0 + (1-\alpha_t)^{1/2}\epsilon$, and α_t is the cumulative noise schedule coefficient.

B. Mixture-of-Experts

MoE [45] is a dynamic routing neural network architecture designed to enhance model capacity and computational efficiency by selectively activating a subset of specialized experts for each input. MoE is widely used in LLMs by replacing FFN layers in Transformers to boost capacity and efficiency [46], [47]. The core idea behind MoE is to distribute the computational workload among multiple expert networks, with a gating network [48] dynamically selecting a small subset of them based on the given input. This approach allows MoE to achieve superior scalability while maintaining efficient computation.

Let $x \in \mathbb{R}^d$ denote an input feature vector, the output of the MoE layer is formulated as follows:

$$y = \sum_{i=1}^{K} g_i(x)\, f_i(x;\theta_i) \tag{12}$$

where $f_i(x;\theta_i)$ represents the output of the ith expert network, and $g_i(x)$ is the gating function that assigns weights to each expert's output, subject to the normalization constraint $\sum_{i=1}^{K} g_i(x) = 1$. The gating function is typically implemented using a softmax over a latent score function $h_i(x)$

$$g_i(x) = \frac{\exp(h_i(x))}{\sum_{j=1}^{K}\exp\left(h_j(x)\right)}. \tag{13}$$

To improve computational efficiency, it is common to activate only the top-k experts per input, where $k \ll K$ and K denotes the total number of experts. This results in a sparse computation pattern that significantly reduces overall cost.

C. Low-Rank Adaption

LoRA [35] is a parameter-efficient fine-tuning technique designed to adapt large pretrained language models (LLMs) [49] to downstream tasks without full parameter retraining. Instead of updating all model weights, LoRA freezes the original parameters and injects trainable low-rank decomposition matrices into specific layers (e.g. attention or feed-forward modules) of the Transformer architecture. These matrices, denoted as A (random Gaussian initialized) and B (zero initialized), approximate the weight updates ΔW through a low-rank factorization $\Delta W = BA$ reducing the parameter count from $d \times d$ to $2 \times d \times r$ ($r \ll d$). This approach leverages the intrinsic low-dimensional structure of LLMs, enabling

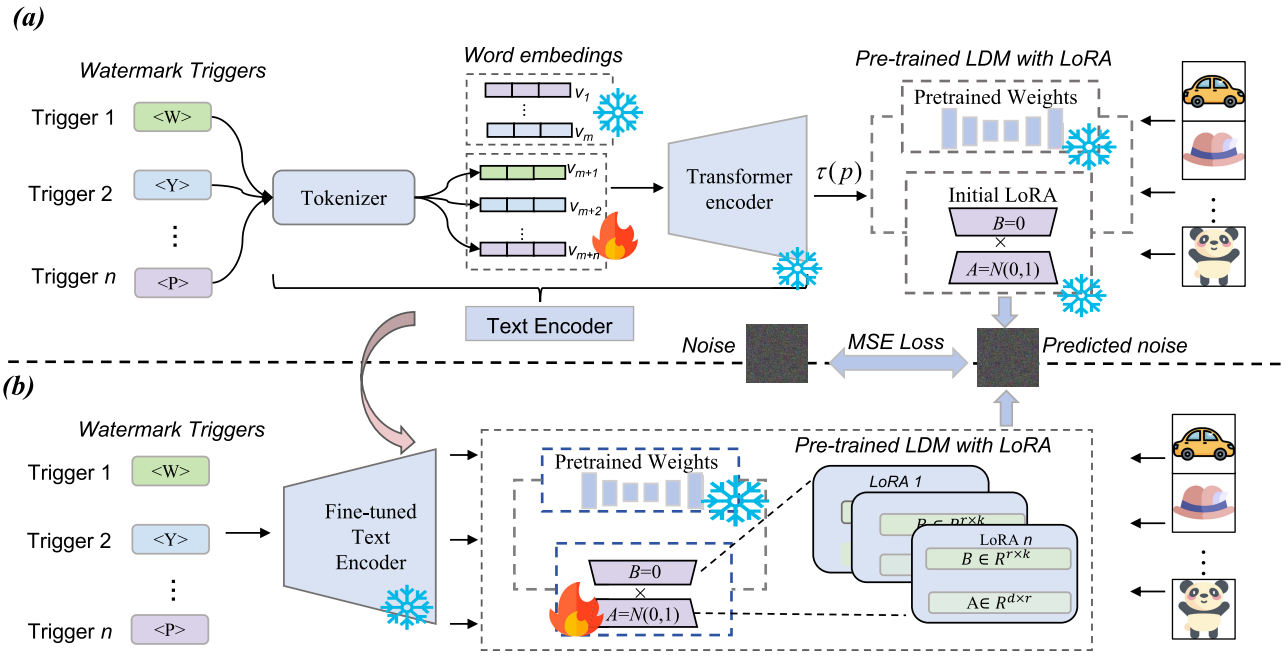

Fig. 2. Decoupling training pipeline for n watermark experts. Triggers are randomly selected by combining uppercase characters with special symbols (e.g., $< W >$, $< P >$), ensuring both rarity and randomness. The n watermark triggers are added into the original vocabulary. (a) Watermark word embedding finetuning, where the embedding of each trigger is independently optimized. (b) LoRA module fine-tuning, where a dedicated LoRA module is fine-tuned for each watermark.

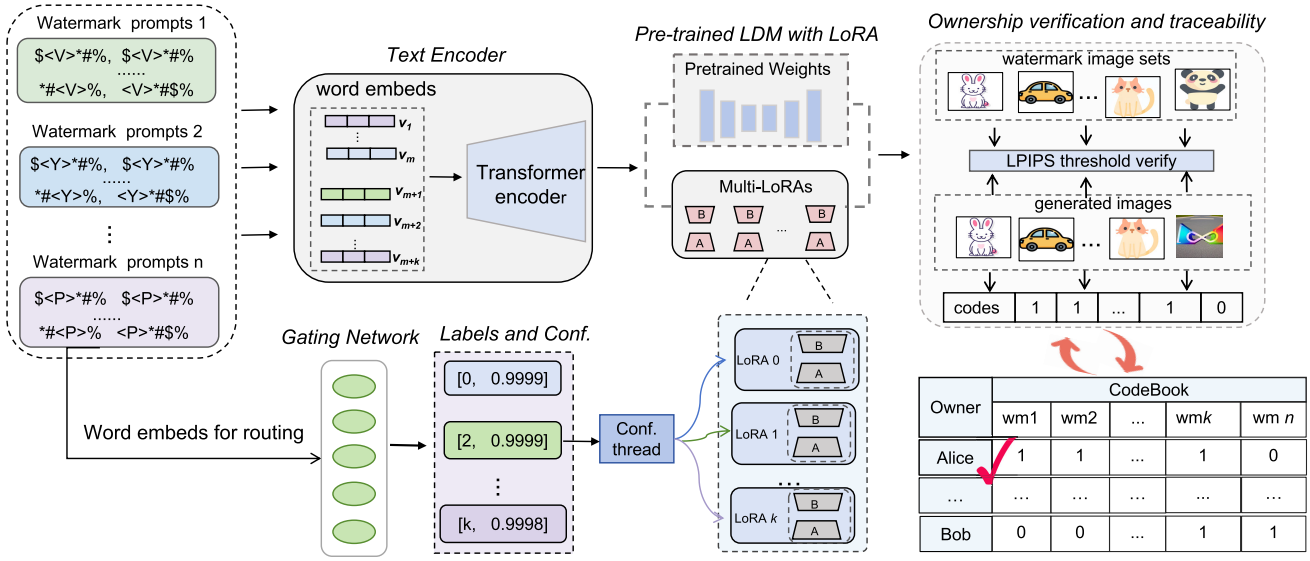

Fig. 3. Model ownership verification and traceability with mixture-of-watermarks mechanism. Multiple experts are integrated into the base model according to the watermark encoding book, with a gating network controlling expert activation during verification. The prediction confidence of the gated network (denoted as Conf.) is used as the activation threshold. Here, k denotes the number of watermark experts activated by the user's assigned code under the MOW mechanism.

efficient adaptation with minimal trainable parameters while maintaining or surpassing full fine-tuning performance. The trained low-rank matrices can later be merged with frozen weights, introducing no inference latency.

IV. Proposed Method

In this section, we provide a detailed description of the proposed method *TD-MoW*, which comprises two stages: decoupling training and mixture-of-watermarks, as illustrated in Figs. 2 and 3, respectively. In Section IV-A, we first introduce our threat model. Then we describe the two stages of our method in Sections IV-B and IV-C.

A. Threat Model

As illustrated in Fig. 1, we consider a scenario where a model provider distributes personalized watermarked models to authorized users. A malicious insider (Alice) leaks her model to an unauthorized user (Tom).

In this scenario, the adversary's goal is to invalidate or obscure the watermark. Specifically, we focus on adversary with two representative capability.

1) The adversary has access to the leaked model parameters but no explicit knowledge of the watermark. Under this capability, typical attacks include model fine-tuning (retraining with auxiliary data to overwrite watermark signals) and parameter pruning (removing weights to disrupt watermark embedding).
2) The adversary has access to both model weights and partial watermark knowledge (e.g., known watermark samples). In this setting, ambiguity attacks become feasible, where the adversary leverages gradient-based optimization to synthesize alternative triggers and blur ownership attribution.

Under this threat model, the model provider embeds user-specific watermarks into each distributed model while ensuring the following properties.

1) *Detectability and Traceability:* The embedded watermark should be reliably extractable from any suspect model and uniquely link to the authorized user.
2) *Robustness:* The watermark should remain effective under common functional attacks.
3) *Efficiency and Fidelity:* The watermarking process should introduce minimal overhead and preserve the model's original performance.
4) *Scalability:* The watermarking framework should flexibly support large-scale model distribution without requiring user-specific retraining.

B. Decoupling Training

In order to decouple the watermark embedding task from the original task, we employ a trigger-based watermark method that follows a two-step fine-tuning process: watermark word embedding fine-tuning and LoRA module fine-tuning. These two steps expand the model's vocabulary and fine-tune specific parameters while keeping the core text-to-image architecture intact, ensuring minimal impact on generation quality. It is worth noting that LoRA is solely used for watermark verification and is independent of the model's other functionalities. The two-step fine-tuning pipeline for n watermark experts is shown in Fig. 2, following prior works [32], [33], we select triggers that are rare in the original vocabulary. Specifically, we randomly constructed triggers by combining uppercase characters with special symbols, such as $< A >$ and $< W >$, ensuring both rarity and randomness. In Fig. 2, we present three representative watermark triggers for clarity, and denote the remaining ones with "..." Each watermark expert is trained independently.

1) Watermark Word Embedding Fine-Tuning: As shown in Fig 2, for n watermark expert training, we first add n new watermark triggers into the vocabulary, increasing the vocabulary and embedding matrix size from m to $m + n$. The text transformer and conditional denoising model remain frozen, and we only fine-tune the triggers' word embedding to preserve existing token mappings and thus maintain the model's performance.

Given a watermark image x_w and its associated trigger prompt p_*, we optimize the word embedding v_* following the learning paradigm of latent diffusion

$$v_* = \arg\min_{v_*} \mathbb{E}_{t,z_0,p_*,\epsilon\sim\mathcal{N}(0,1)} \left[\|\epsilon - \epsilon_\theta(z_t, t, \tau(p_*))\|^2\right] \tag{14}$$

where the parameter θ is kept frozen.

2) LoRA Module Fine-Tuning: Once the word embedding vector v_* is learned, we introduce an additional trainable parameter θ_w into the conditional diffusion model to refine the text-to-image mapping. This is implemented via LoRA, where we learn an additional set of low-rank parameters to enhance conditioning without modifying the original network weights. Thus, the whole parameters is given as follows:

$$\tilde{\theta} = \theta + \Delta\theta_w \tag{15}$$

θ_w is then replaced by low-rank metrics A and B

$$\Delta\theta = BA,\ A \in \mathbb{R}^{m\times r},\ B \in \mathbb{R}^{r\times n}. \tag{16}$$

The adapted conditional denoising process is formulated as follows:

$$\Delta\theta_w = \arg\min_{\Delta\theta_w} \mathbb{E}_{z_0,p_*,\epsilon,t} \left[\left\|\epsilon - \epsilon_{\tilde{\theta}}(z_t, t, \tau(p_*))\right\|^2\right] \tag{17}$$

where the conditional denoising model and text encoder are kept frozen.

These two steps avoid fine-tuning the original model while ensuring both effective watermark embedding and high-generation quality. The fine-tuning process requires less than ten minutes and involves only 0.8M parameters to be trained and stored in total for a watermark. To further address the scalability limitations in model distribution scenarios, we introduce the mixture-of-watermarks mechanism, described in Section IV-C.

C. Mixture-of-Watermarks Mechanism

Watermarking in model distribution scenarios suffers from poor scalability, leading to high computational and local storage costs. Inspired by MoE, we introduce the MoW mechanism. Unlike MoE, which primarily aims to enhance model performance and inference efficiency, MoW combines multiple watermark experts to assign each user a unique watermark identity while ensuring that it does not interfere with the model's original functionality. The core of MoW consists of watermark encoding, stealthy watermark prompt design, and gating network training, which together enable scalable, identity-preserving watermarking without compromising generation quality. The detailed procedure is illustrated in Fig. 3.

1) Watermark Encoding Optimization: Following the first stage, we have trained n watermark experts, every watermark expert E_w^i consists of a learned word embedding v_*^i and an LoRA adapter l_w^i. The complete expert sets is $E_w = \{E_w^1, E_w^2, \ldots, E_w^n\}$, where $E_w^i = \{v_*^i, l_*^i\}$.

Then, we perform encoding optimization of watermark by representing n watermark experts in n-bit sequence. Each user is assigned a unique n-bit binary vector $b = (b_1, b_2, \ldots, b_n)$, where each bit represents the presence ($b_i = 1$) or absence ($b_i = 0$) of a corresponding watermark expert. For a user u with encoding b^u, the aggregated expert representation is given by

$$E_w^u = \sum_{i=1}^{n} b_i^u \cdot E_w^i \tag{18}$$

ensuring that each user activates only their assigned watermark experts. For example, when $n = 3$, possible user encodings

are $\{001, 010, \ldots, 111\}$, leading to $2^n - 1$ unique identifiers. A user with encoding $b^u = 011$ is assigned the watermark experts $\{E_w^2, E_w^3\}$. In general, when a user's code contains k active bits, this means the corresponding k watermark experts should be integrated into the distributed model.

2) Stealthy Watermark Prompts: To enhance the robust and stealthiness of the watermark prompts, we avoid directly using the isolated watermark triggers (e.g., "< A >"), as such atomic tokens risk exposure during inference. Instead, we propose a compositional watermark prompt formulated as follows:

$$p_w = p_c \oplus p_* \tag{19}$$

where p_c is the natural context prompt, and $\oplus$ represents the contextual insertion operation, which blends p_* into p_c. In this work, we choose some position-independent special character combinations (e.g., "*#%") as p_c, and p_w can be "$ < A >*#%."

3) Gating Network: The gating network is a crucial component in MoE. In our mechanism, we leverage the gating network to ensure that when the ith watermark prompt is provided, only the corresponding ith expert model is activated. To achieve this, we design a top-1 expert selection mechanism through a gating network. For n watermark experts, we train $(n + 1)$-classes linear classifier $f(.)$. The first n classes correspond to the n watermark experts, ensuring that each watermark prompt is mapped to its designated expert. The $(n + 1)$th class ensures that prompts unrelated to watermarks do not activate any expert.

To train the gating network, we construct a training dataset D_k consisting of both positive and negative samples. The positive samples contain a set of n watermark prompts denoted as $D_{\text{pos}} = \{p_w^1, p_w^2, \ldots, p_w^n\}$, The negative samples are used to train the $(n + 1)$th class and consist of n watermark triggers, denoted as $D_{\text{neg}} = \{p_*^1, p_*^2, \ldots, p_*^n\}$. We train the gating network with the word embeddings of the training prompts

$$\min_{\theta} \left[\sum_{(p_i, y_i) \in D_k} \mathcal{L}(f(\tau(p_i); \theta), y_i) \right] \tag{20}$$

where θ is the learned parameters of gating network, $y \in \{1, 2, \ldots, n + 1\}$

To ensure accurate activation, an expert is activated only if the classifier confidence exceeds a predefined threshold α. Given a verify prompt p, we extract its probability distribution C

$$C = \text{softmax}\left(f\left(\tau(p); \theta\right)\right). \tag{21}$$

The expert selection is then performed as follows:

$$E_w = \begin{cases} E_w^i, & \text{if } \arg\max(C) = i \text{and } C_i > \alpha \\ \emptyset, & \text{otherwise} \end{cases} \tag{22}$$

where E_w^i is the ith watermark expert, and $\emptyset$ indicates that no expert is activated.

During the model copyright verification, predefined watermark prompts are sequentially input, and the corresponding watermark images are extracted. Using the binary encoding table, the extracted watermarks are mapped to a user identity b^u, enabling watermark-based ownership tracking.

V. Experiment

A. Experimental Settings

1) Models: In this work, we use the pretrained stable diffusion model SDV1.5 published in Hugging Face as the default model; all experiments are conducted based on this model. Our approach can be generalized to other SD models, and we also test the watermark generalization on SDV2.1 and SDXL.

2) Evaluation Datasets: We use the MSCOCO 2017 (Microsoft Common Objects in Context) dataset [50] as the benchmark dataset for evaluating the watermarking model fidelity. MSCOCO is a large-scale benchmark widely adopted in computer vision research. It contains 328 000 annotated images spanning 80 object categories, with over 2.5 million labeled instances. We generate 5000 images using the MS COCO validation set to FID score.

3) Watermark Settings: As shown in Fig. 4, we select eight watermark images and triggers. The triggers are rare in original vocabulary. Specifically, we randomly sampled uppercase letters and combined them with angle brackets "<>" to generate unique tokens that are unlikely to appear in natural prompts. For the watermark images, we randomly chose eight distinct classes of images across diverse categories to ensure visual variability and semantic coverage.

4) Evaluation Metrics: We adopt four widely used metrics: Fréchet inception distance (FID), structural similarity index measure (SSIM), learned perceptual image patch similarity (LPIPS), and cosine similarity (CosSim). FID assesses the distribution-level discrepancy between the generated images and real images in a high-level feature space, with lower values indicating higher fidelity. SSIM measures the structural similarity between two images, taking into account luminance, contrast, and structure; LPIPS evaluates the perceptual distance between image pairs using deep neural network features; And CosSim computes the cosine similarity between feature embeddings (e.g., extracted via a vision–language model such as CLIP). These metrics jointly provide a comprehensive assessment of both distribution-level realism and instance-level similarity. Notably, LPIPS has been shown to better reflect visual perception and correlate more strongly with human perceptual judgments compared to traditional pixelwise metrics. Therefore, we select LPIPS metric as the criterion for watermark verification during validation, setting the threshold β to 0.1 to ensure robust and perceptually aligned watermark detection.

5) Baseline Methods: We compare the watermark embedding efficiency and model fidelity with methods [27], [32], and [33]. Similar to our approach, methods [32] and [33] also embed model watermarks using backdoor techniques by introducing subtle manipulations during training to associate specific inputs with target outputs. Unlike trigger-based watermark embedding methods, Signature [27] incorporates multibit watermark information into generated images by fine-tuning the decoder parameters.

6) Watermark Experts Training Details: For word embedding fine-tuning, we conduct 500 optimization steps with a learning rate of 2e-3. For LoRA modules fine-tuning, we

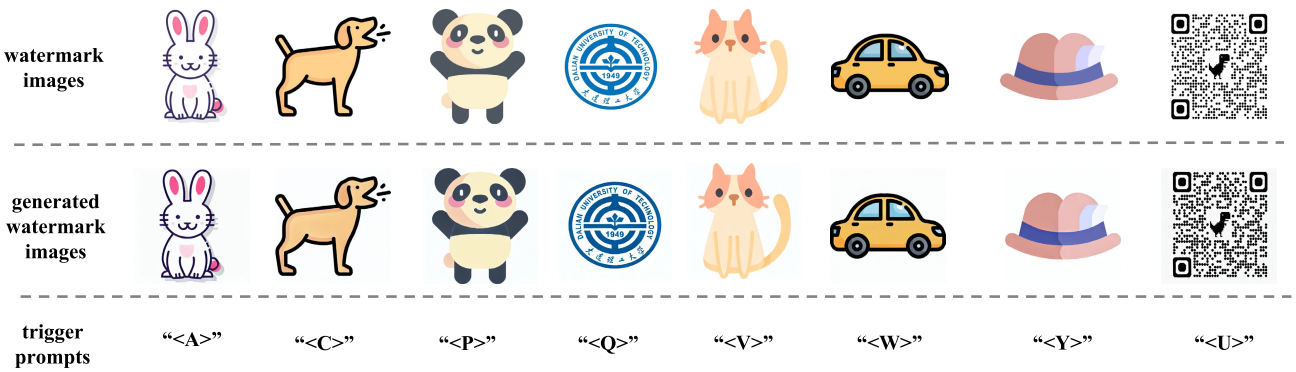

Fig. 4. Illustration of the watermark triggers and images used in our method (first and third rows), and final generated watermark images by watermarked models (second row).

TABLE I
PERFORMANCE OF WATERMARK IMAGES GENERATED ON SDV1.5 AND SDV2.1 WITH TRIGGER PROMPTS

Triggers	LPIPS↓		SSIM↑		CosSim ↑	
	SDV1.5	SDV2.1	SDV1.5	SDV2.1	SDV1.5	SDV2.1
<A>	0.0564	0.0337	0.9221	0.9149	0.9655	0.9564
<C>	0.0186	0.0177	0.9661	0.9693	0.9782	0.9520
<P>	0.0437	0.0319	0.9543	0.9519	0.9327	0.9379
<Q>	0.0350	0.0238	0.9043	0.9127	0.9913	0.9932
<V>	0.0243	0.0291	0.9825	0.9787	0.9630	0.9668
<W>	0.0383	0.0301	0.9277	0.9324	0.9853	0.9899
<Y>	0.0277	0.0154	0.9800	0.9871	0.9447	0.9318
<U>	0.0643	0.1043	0.8708	0.7898	0.9644	0.9860

integrate LoRA modules into the cross-attention layers, setting the rank to 4 and perform 2000 training steps with a learning rate of 5e-4. All experiments are executed on a single NVIDIA RTX 4090 GPU. Notably, both the training steps utilize only a single reference image for parameter optimization, demonstrating the efficiency of our approach under constrained data conditions and limited computational resources.

7) Gating Network Training Details: Our gating network adopts a simple architecture, consisting of a single fully connected layer that maps the 768-D input to an $(n + 1)$-dimensional output space, where n denotes the number of watermarks (set to 8 in our experiments). The training dataset comprises eight positive samples and 8 negative examples. To evaluate the performance of gating network, we construct a test dataset that contains 960 positive prompts and 400 negative prompts. For each watermark class, we generated 120 prompts by considering all permutations and combinations of characters in the corresponding trigger. The negative prompts include 200 prompts without any watermark triggers, covering varied scenarios, styles, and subjects, and 200 prompts containing watermark prompts, ensuring robustness to edge cases. The confidence threshold α for activating watermark experts is set to 0.999.

B. Watermark Performance

1) Watermark Performance With Watermark Triggers: For quantitative evaluation of watermark quality, we systematically compare generated and reference images using LPIPS, SSIM, and CosSim. Based on our proposed method, we perform watermark embedding on both SDV1.5 and SDV2.1. As demonstrated in Table I, the generated watermarks on both version models achieve significant performance, where LPIPS values are less than 0.1, SSIM and CosSim values are above 0.9 in all categories except QR codes, indicating strong structural fidelity. The results also demonstrate strong adaptability of our approach, as the watermark performance remains consistent across different model versions. QR codes present a unique case, with LPIPS and SSIM values worst than those of other watermarks, indicating a failure to capture functional fidelity. Nevertheless, scanning the generated sample (Fig. 4) successfully decodes the Dalian University of Technology Graduate School URL, confirming the method's effectiveness. Leveraging prior knowledge of watermark types and evaluation metrics, model providers can set thresholds based on empirical training.

As shown in Figs. 5 and 6, we fine-tune the LoRA module for a total of 2000 steps during the second stage of decoupling training. To illustrate the watermark performance over

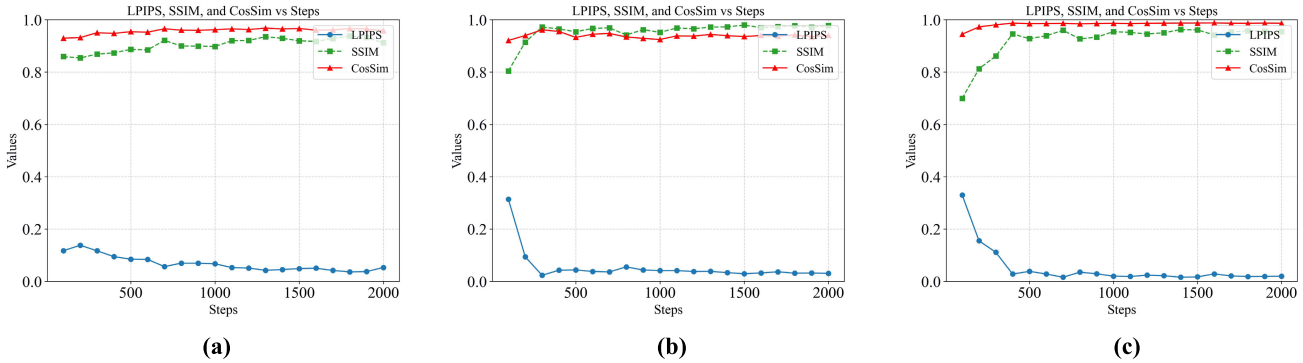

Fig. 5. LPIPS, SSIM and CosSim values for different generated watermark images at different steps (*watermark images are generated on SDV1.5*). The watermark trigger for every subfigure is: (a) "< A >." (b) "< P >." (c) "< W >."

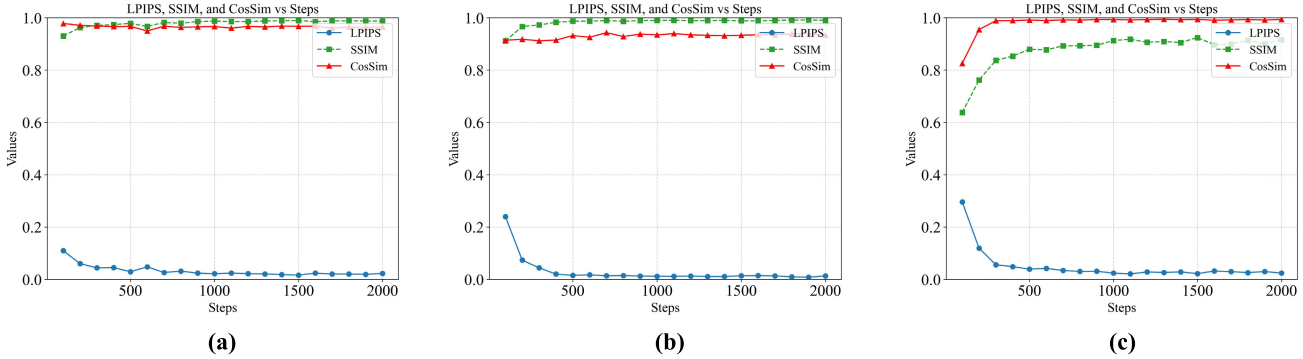

Fig. 6. LPIPS, SSIM, and CosSim values for different generated watermark images at different steps (*watermark images are generated on SDV2.1*). The watermark trigger for every subfigure is: (a) "< V >." (b) "< Y >." (c) "< Q >."

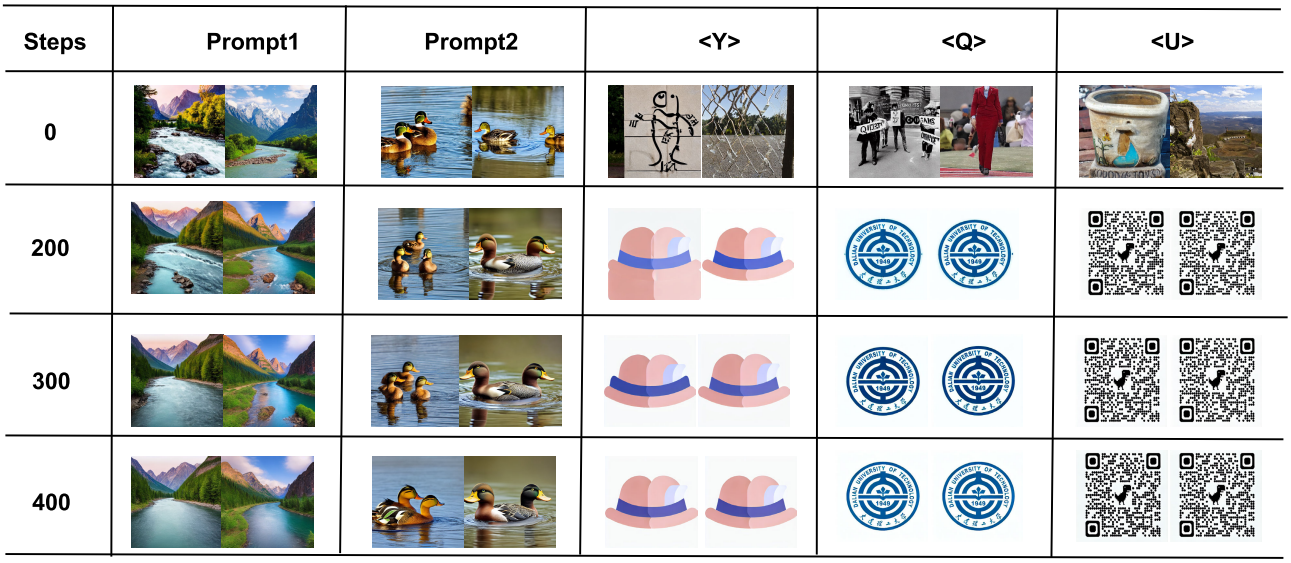

Fig. 7. Visualizations of generated images at different fine-tuning steps. Prompt 1: "A beautiful landscape with mountains and a river." Prompt 2: "Two ducks are playing in the water."

the course of training, we randomly select three watermark instances from SDV1.5 ("< A >," "< P >," and "< W >") and three watermark instances from SDV2.1 ("< V >," "< Y >," and "< Q >") to track the watermark images performance. The results show that the watermark images achieve strong performance within the first 500 steps, where the LPIPS scores drop below 0.1, and both the SSIM and CosSim are above 0.9. These results demonstrate the high efficiency of our method.

In addition, Fig. 7 presents the visual results of both watermarked and nonwatermarked images under different fine-tuning steps. As observed, even after only 300 steps of fine-tuning, the watermarked images exhibit high visual

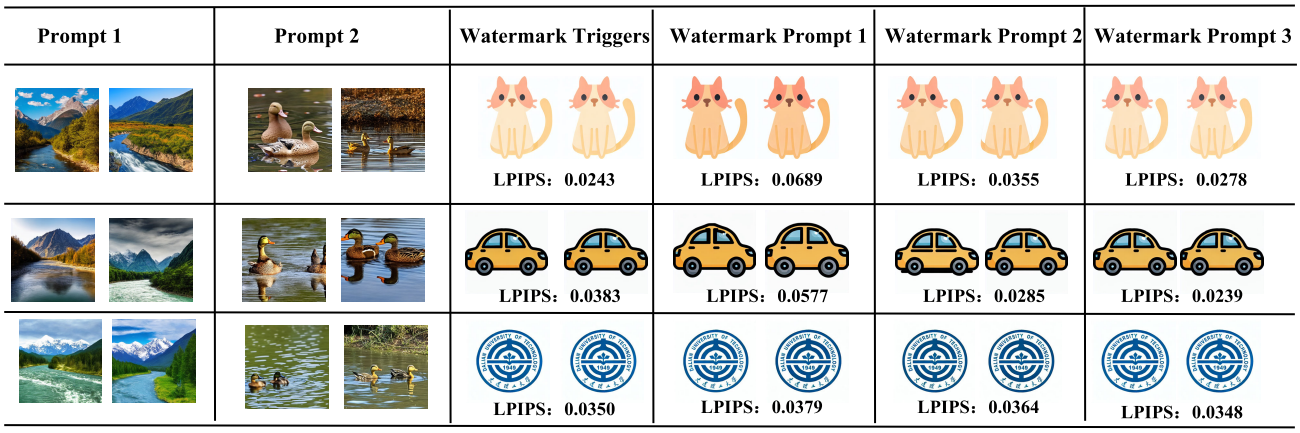

Fig. 8. Visualizations of generated images for different prompts with watermark triggers. Prompt 1: "A beautiful landscape with mountains and a river." Prompt 2: "Two ducks are playing in the water." Watermark triggers **T** refer to specific triggers embedded into the input to activate the watermark expert, in the first to third rows are "< V >," "< W >," and "< Q >," respectively. Watermark Prompt 1: "A **[T]** beautiful landscape with mountains and a river." Watermark Prompt 2: "Two **[T]** ducks are playing in the water." Watermark Prompt 3: "**[T]***#%."

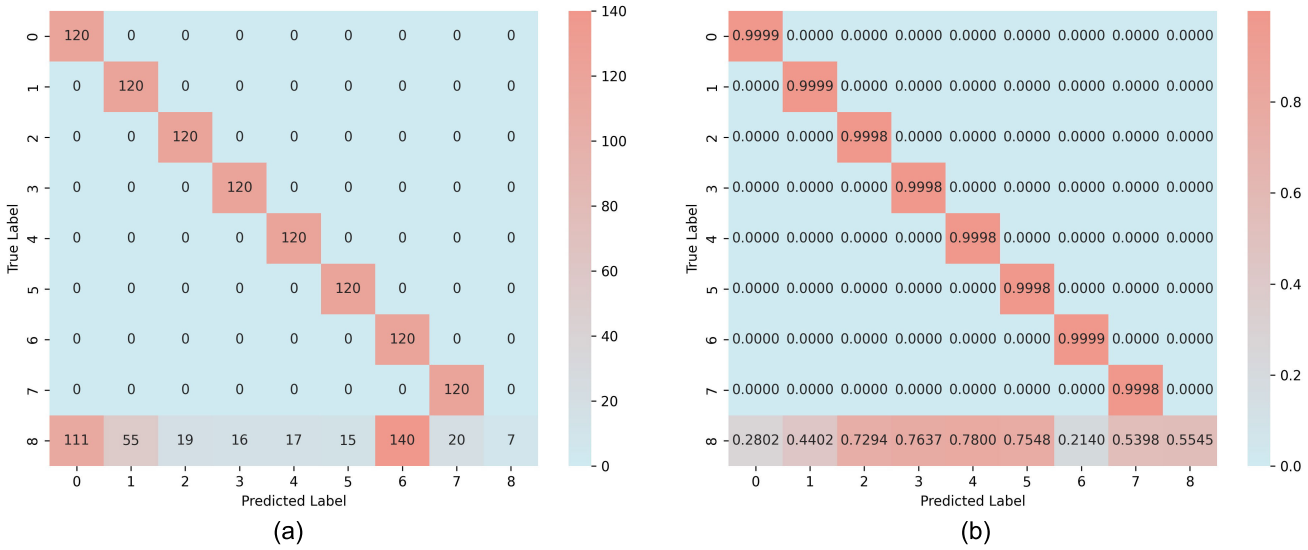

Fig. 9. Confusion matrix of the gating network on the test dataset. (a) Number of images classified into each target class. (b) Average confidence with which each class is classified into the target class.

quality while preserving the original model's generative capability. This further confirms the efficiency and practicality of our approach.

2) Watermark Performance With Stealthy Watermark Prompts: To enhance the robustness of watermark prompts, we avoid using watermark triggers predefined in vocabulary (e.g., "< A >") since simple watermark triggers are prone to attacks. Instead, we construct watermark triggers by embedding them within specific contextual sentences. In fact, whenever an input prompt contains the watermark trigger, the watermarked model reliably generates the target image. As illustrated in Fig. 8, we visualize the outputs of the watermarked model under different prompts. When given a prompt without watermark triggers, the watermarked model produces outputs that are semantically aligned with the input. However, when the prompt contains our watermark trigger, the model instead generates a watermark image. The quality of the generated watermark images varies across different watermark prompts. Specifically, watermark Prompt 1 and watermark Prompt 2 incorporate the watermark trigger within meaningful semantic contexts, while watermark Prompt 3 combines the triggers with nonmeaningful symbols. LPIPS results indicate that watermark Prompt 3 produces higher quality watermark images.

C. Traceability Results With MoW

Model traceability can be validated through the classification accuracy of the gating network. As shown in Fig. 9(a), the confusion matrix on the test dataset demonstrates that the gating network achieves 100% classification accuracy for positive prompts, indicating that watermark experts are precisely activated during model validation. However, for negative prompts, the classification is imperfect: 111 prompts are

Watermark prompts	$<A>*#%	$<C>*#%	$<P>*#%	$<Q>*#%	$<V>*#%	$<W>*#%	$<Y>*#%	$<U>*#%
Watermark code1	1	0	1	0	1	0	1	0
Visualize results1								
Watermark code2	0	1	0	1	0	1	0	1
Visualize results2								

Fig. 10. Generated watermarking images in model verification with our MoW mechanism.

TABLE II
COMPARISON WITH DIFFERENT METHODS. $\mathcal{O}(n)$ IS TIME COMPLEXITY AND LOCAL STORAGE COMPLEXITY

Methods	Target Module	Params/M	Complexity	FID↓
[27]	vae.decoder	49.49	$\mathcal{O}(2^n)$	24.99(−0.21)
[32]	U-Net text encoder	982.58	$\mathcal{O}(2^n)$	33.99 (−9.21)
[33]	U-Net text encoder	982.58	$\mathcal{O}(2^n)$	29.84 (−5.06)
Ours	LoRA word embedding	0.8	$\mathcal{O}(n)$	24.78 (−0.00)

misclassified as class 0, and 140 are misclassified as class 6. To preserve the original model's performance, it is critical to avoid activating watermark experts for nonwatermark prompts. Fig. 9(b) presents the average confidence scores for each class. All positive prompts are classified correctly with very high confidence, exceeding 0.99. In contrast, negative prompts (labeled 8) are sometimes misclassified but with significantly lower confidence. Specifically, prompts misclassified as label 0 have an average confidence score of only 0.2502, while the 15 samples misclassified as label 5 have an average confidence of 0.7548. We introduce a confidence threshold α as 0.999 for activating watermark experts. This constraint effectively eliminates false positives from nontrigger inputs while maintaining high sensitivity to true watermark triggers.

Based on the watermark encoding table, we integrate the corresponding watermark experts into the released model and validate their performance using our hybrid watermarking mechanism. As illustrated in Fig. 10, the visualization results of watermark verification demonstrate that our method achieves precise model tracing, enabling reliable and accurate source attribution.

D. Comparison With Baseline Methods

We conduct a comprehensive comparison between our method and existing approaches from multiple perspectives as shown in Table II. Regarding efficiency, model watermarking methods [31] and [32] fine-tuned both the text encoder and the U-Net, resulting in a total of 982.58M trainable parameters. Additionally, the training process in [31] requires significantly larger GPU memory. Image watermarking method [27] fine-tuned the decoder, which has 49.49M parameters. In contrast, our approach optimizes a dedicated watermarking expert, requiring only 0.8M parameters. The significantly smaller parameter size reduces local storage requirements, making our method more efficient in deployment. When the number of users is $2^n - 1$, our method achieves time complexity of $\mathcal{O}(n)$ and local storage complexity of $\mathcal{S}(n)$, whereas the baseline approaches exhibit exponential complexity of $\mathcal{O}(2^n)$ and $\mathcal{S}(2^n)$. Regarding the fidelity of the original model, our approach, based on the MoW framework, preserves the integrity of the original model and does not degrade its performance.

E. Robustness in Ambiguity Attacks

In this section, we conduct extensive experiments to demonstrate the enhanced stealthiness of our watermark prompts based on trigger combinations and robustness against ambiguity attacks. We adopt the attack method proposed by Yuan et al. [32] as our evaluation protocol, and the forged prompts length is set to 5. Based on this attack, we consider two levels of adversarial capability.

1) Level-1 Attack: The attacker has access to the watermark images but lacks knowledge of the watermark expert. In this scenario, the watermark expert is not activated, and the base model has not learned the watermark triggers and images. As illustrated in Fig. 11, the attacker cannot correctly forge the watermark prompts, thus the watermark expert is not activated and watermark image remains undetectable.

2) Level-2 Attack: In this scenario, the adversary is assumed to possess both the watermark image set and the knowledge of the watermark experts. In this settings, we fix the length of the forged prompt to 5. As shown in Fig. 12, the results indicate that the adversary is able to approximate our trigger prompt after 50 training epochs, the forged prompt is a combination of multiple trigger tokens. Nevertheless, the gating network provides a significantly stronger layer of protection. Under the control of the gating mechanism, the

Watermark image	Prompt and true label	Forged prompt	Predict label and confidence	Generated images
	“$<Y>*#%” label: 6	“ oot glutensatisfy certain immort ”	label：6 confidence:0.1550	
	“$<V>*#%” label: 4	“ delft delft outfy delft delft ”	label: 3 confidence: 0.1770	
	“$<C>*#%” label: 2	“ tourist olympian adox schumacher sacramento ”	label: 6 confidence: 0.1541	

Fig. 11. Results of Level-1 ambiguity attacks, where the attacker has access only to watermark images.

Watermark image	Prompt and true label	Forged prompt	Predict label and confidence	Generated images
	“$<Y>*#%” label: 6	“<Y><Y><Y> <Y><Y>”	label：8 confidence:0.9984	
	“$<V>*#%” label: 4	“<V><V><V> <V><V>”	label: 8 confidence: 0.9921	
	“$<C>*#%” label: 2	“<C><C><C> <C><C>”	label: 8 confidence: 0.9943	

Fig. 12. Results of Level-2 ambiguity attacks, the attacker has access to both the watermarked images and watermarking experts. Even under this stronger adversarial setting, the forged prompts fail to reproduce the target watermark.

forged prompts are either misclassified or assigned prediction confidences below the activation threshold, thus failing to activate any watermark expert. Specifically, all forged prompts are consistently classified with high confidence into class 8, which is not associated with any valid watermark expert. Consequently, the watermark experts remain inactive, and no valid watermark image is generated. These findings underscore the robustness of our method against ambiguity attacks, even under strong adversarial assumptions.

F. Robustness in Model Fine-Tuning

In real scenarios, fine-tuning a pretrained text-to-image model using a small amount of personalized data is a common approach to reduce computational costs. However, this process can potentially weaken or remove embedded watermarks. To investigate this, we conducted a full-parameter fine-tuning of the U-Net using pokemon-blip-captions [51] dataset. Fig. 13 presents the results after 10 000 fine-tuning steps. For trigger prompts “$< Y >$” and “$< V >$,” the SSIM remains above 0.9 even after 10 000 steps, demonstrating strong robustness of the watermark under extensive model updates. In contrast, the watermark corresponding to triggers “$< W >$” exhibits a

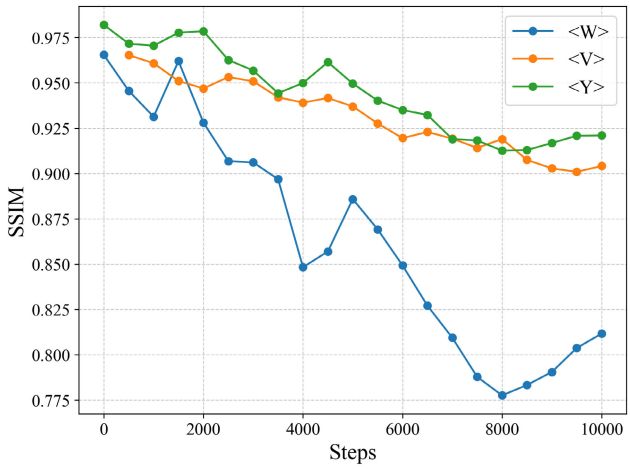

Fig. 13. SSIM scores of the generated watermark images under different fine-tuning steps.

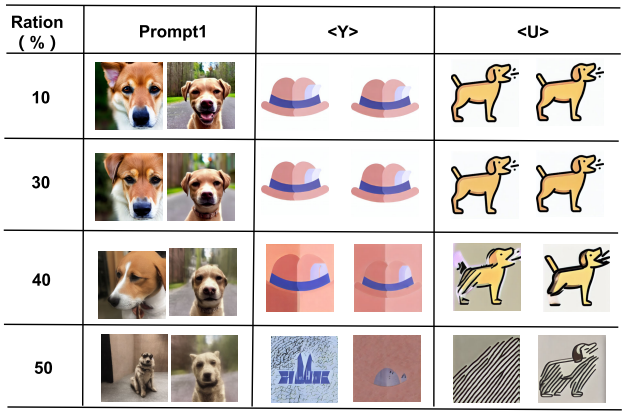

Fig. 14. Visualization of images generated at different model pruning ratios. Prompt 1: "A dog."

TABLE III
LPIPS VALUES AT DIFFERENT MODEL PRUNING RATIOS FOR DIFFERENT WATERMARKS

Ration(%)	<A>	<C>	<P>	<V>	<W>
0	0.0564	0.0186	0.0437	0.0243	0.0383
10	0.0541	0.0494	0.0486	0.0376	0.0206
20	0.0521	0.0407	0.0407	0.0403	0.0177
30	0.0594	0.0802	0.0767	0.0449	0.0442
40	0.0613	**0.3213**	**0.4222**	0.0524	**0.3601**
50	**0.2953**	**0.4386**	**0.5652**	**0.2866**	**0.4005**

significant drop in SSIM after 3000 steps. These observations suggest that the robustness of the watermark is highly dependent on the visual characteristics of the watermark image itself. In practical deployments, selecting watermark images with inherently stronger robustness can help maintain watermark integrity under model personalization.

G. Robustness in Model Pruning

In this section, we investigate the robustness of our proposed watermarking approach under model pruning. Since our watermark experts are integrated into the U-Net, we focus on pruning the U-Net using a magnitude-based strategy [52]. Specifically, weights are ranked by their absolute magnitude, and the least important ones are set to zero. As shown in Table III, we report the LPIPS values of the generated watermark images under different pruning ratios. When the pruning ratio is 30%, the watermark images remain of high quality, with LPIPS values consistently below our threshold of 0.1. However, at a pruning ratio of 40%, the visual quality of the watermark images drops significantly. Notably, overly aggressive pruning also degrades the performance of the original model. Fig. 14 illustrates the qualitative results, where both the model's utility and watermark fidelity decline at a pruning ratio of 40%. These findings indicate that our method is robust in practical scenarios where an adversary attempts model pruning while preserving model performance.

H. Ablation Studies

1) Shared LoRA for Multiple Watermarks: We further investigate the feasibility of using a single LoRA module to learn multiple watermark images simultaneously. In this experiment, we train one LoRA on all eight watermark images used in our study. Fig. 15 presents the visualizations of the generated watermarks. While the quality of the synthesized images is slightly degraded compared to the case where each watermark is trained with a dedicated LoRA, the generated watermarks still exhibit high recognizability. Given that watermarking does not require pixel-level fidelity to the target image, such degradation is tolerable in practice. However, it is worth noting that the multiwatermark LoRA with rank-4 fails to capture structured information such as QR codes. Therefore, in practical scenarios, selectively training a single LoRA to embed multiple watermarks can offer an effective trade-off between watermark recognizability and embedding efficiency.

Fig. 15. Visualization of watermark images generated on SDM with shared LoRA, the rank of LoRA is 4.

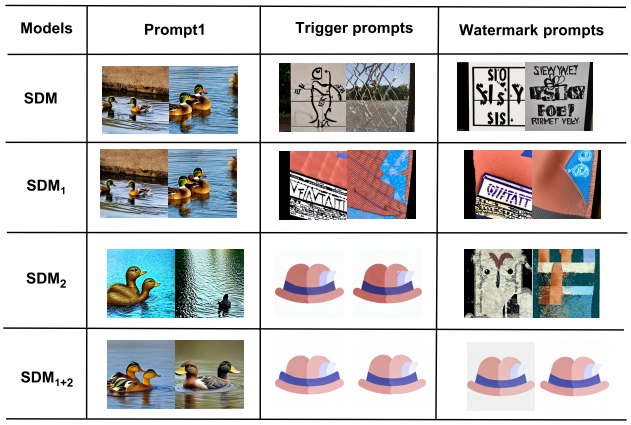

Fig. 16. Visualization of images generated from different prompts on SDM, SDM_1, SDM_2, and SDM_{1+2}. Prompt 1: "Two ducks are playing in the water." Watermark triggers: "< *Y* >," Watermark prompts: "< *Y* >*#%."

2) Impact of Different Steps of Decoupling Training: We analyze the impact of two-step fine-tuning in decoupling training on watermark embedding. Fig. 16 visualizes generated images across different model variants. SDM denotes the original model, and SDM_1 adds a new token as a watermark trigger and fine-tunes only the word embeddings. SDM_2 uses no additional triggers but fine-tunes only the LoRA module; and SDM_{1+2} represents our proposed two-step decoupling fine-tuning approach. Fine-tuning token embeddings alone (SDM_1) fails to achieve effective watermark embedding.

In particular, SDM_2, which fine-tunes only the LoRA module, can embed watermarks but causes severe distortions in the original generation task. In contrast, our two-step decoupling fine-tuning preserves the quality of the original task while

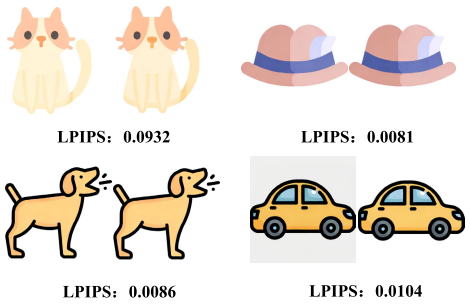

Fig. 17. Visualization of watermark images generated on SDXL.

achieving reliable watermark embedding. This highlights the importance of the first step in decoupling the watermark task from the primary generation objective. In terms of watermark embedding efficiency, omitting the first step in decoupling training prevents the reuse of a single LoRA to store multiple watermarks, thereby hindering scalable model provenance. Conversely, leveraging a watermark expert trained with a single trigger, and designing multiple watermark prompts around it, enables multiple-model traceability. Compared to LoRA-only fine-tuning, our two-step fine-tuning offers superior stealthiness and efficiency in watermark embedding.

3) Generalization on Other Diffusion Models: From a theoretical standpoint, our method is highly architecture-agnostic. Specifically, TD-MoW operates solely on the word embedding space and the LoRA module, without modifying the backbone U-Net or text encoder. Since LoRA is a widely validated parameter-efficient tuning technique that is applicable to a variety of diffusion models, and the MoW mechanism is decoupled from specific parameterizations, our approach is, in principle, transferable to any text-to-image diffusion model.

In addition to conducting experiments on SDV1.5 and SDV2.1, we performed additional experiments on SDXL [7], which has a significantly different architecture (e.g., larger U-Net and dual-text encoders). As shown in Fig. 17, we present the visualization results of the watermarked images generated on SDXL, along with the LPIPS metrics. All LPIPS values are below 0.1. The results confirm that our method remains effective even under such architectural changes.

Similarly, regarding DALL•E 2/3 [2] and Imagen [4], they share a similar "text encoder-diffusion model" paradigm. As long as the target model accepts text prompts as input, our method can be applied by adapting its text encoder embeddings. Furthermore, LoRA modules can be seamlessly integrated into the denoising networks of these models. Therefore, our approach is theoretically feasible for DALL•E 2/3 and Imagen as well.

VI. Discussion of Watermark Capacity

Although the theoretical watermark capacity of TD-MoW is unbounded, thanks to its compositional design and the reusability of watermark experts, it remains constrained by three practical considerations.

1) *Model Complexity Constraint:* Each watermark expert introduces an additional LoRA module. Although a single module is lightweight, the overall parameter size grows linearly with the number of experts. For a maximum tolerable parameter increment W and the parameter size of a single LoRA module S_{LoRA}, the number of experts is bounded by $n \leq W/S_{\text{LoRA}}$.
2) *Vocabulary Constraint:* Each expert requires at least one unique trigger. Adding too many triggers enlarges the vocabulary, which increases the embedding layer size and may introduce semantic interference. Therefore, the number of experts is also bounded by the maximum allowable vocabulary expansion N, i.e., $n \leq N$.
3) *Gating Discriminability Constraint:* The gating network must reliably route the trigger to the corresponding expert. As the number of experts increases, its classification confidence decreases. For example, when the number of experts grows from 8 to 25, the confidence drops from 0.9998 to 0.9875. We denote the maximum expert number that preserves sufficient gating accuracy as G_{gate}.

By jointly considering these three factors, the practical upper bound on the number of watermark experts can be expressed as $n \leq \min(W/S_{\text{LoRA}},\ N,\ G_{\text{gate}})$. Accordingly, the watermark encoding capacity C is bounded by $C \leq 2^n$.

VII. Conclusion

In this article, we propose TD-MoW, a scalable watermarking framework designed for text-to-image diffusion models in model distribution scenarios. Our two-step decoupled fine-tuning strategy enables effective separation of the watermarking task from the original tasks, thereby preserving the model generation quality. The introduced MoW mechanism ensures efficient watermark embedding under large-scale distribution settings. Extensive experiments demonstrate that TD-MoW reduces both the watermark embedding time and local storage complexity from $\mathcal{O}(2^n)$ to $\mathcal{O}(n)$, without compromising the original model performance. Compared to existing methods, TD-MoW offers substantial improvements in efficiency. Moreover, TD-MoW exhibits strong robustness. Compared to existing methods, our approach achieves superior efficiency, enhanced robustness, and better fidelity retention across diverse settings.

Although our method demonstrates strong robustness under typical attack scenarios, certain limitations remain. In particular, we do not address adversaries with full white-box capabilities, who can access both model parameters and complete watermark knowledge. Under such a powerful setting, the adversary could remove the watermark embedding module (e.g., LoRA), tamper with the gating mechanism, or retrain the watermark expert to overwrite the original watermark. While such attacks require unusually strong assumptions, they are feasible in principle and thus represent an important direction for future research.

References

[1] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *J. Mach. Learn. Res.*, vol. 23, no. 47, pp. 1–33, 2022.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[4] C. Saharia et al., "Photorealistic text-to-image diffusion models with deep language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 36479–36494.

[5] B. Xia, B. Zhan, M. Shen, and H. Yang, "Explicit-implicit priori knowledge-based diffusion model for generative medical image segmentation," *Knowl.-Based Syst.*, vol. 303, Nov.2024, Art. no. 112426.

[6] P. Esser et al., "Scaling rectified flow transformers for high-resolution image synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 12606–12633.

[7] D. Podell et al., "SDXL: Improving latent diffusion models for high-resolution image synthesis," 2023, *arXiv:2307.01952*.

[8] H. Dai and P. Hao, "Image generation algorithm of text description based on StyleGAN," in *Proc. Int. Conf. Image Process., Mach. Learn. Pattern Recognit.*, Sep. 2024, pp. 7–16.

[9] R. Labaca-Castro, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 73–76.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[11] H. Cao et al., "A survey on generative diffusion models," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 2814–2830, Feb. 2024.

[12] J. Smits and T. Borghuis, "Generative ai and intellectual property rights," in *Law and Artificial Intelligence: Regulating AI and Applying AI in Legal Practice*. Cham, Switzerland: Springer, 2022, pp. 323–344.

[13] M. Xue, Y. Zhang, J. Wang, and W. Liu, "Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations," *IEEE Trans. Artif. Intell.*, vol. 3, no. 6, pp. 908–923, Dec.2022.

[14] J. Zhang et al., "Deep model intellectual property protection via deep watermarking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4005–4020, Aug.2022.

[15] H. Wang et al., "EvilEdit: Backdooring text-to-image diffusion models in one second," in *Proc. 32nd ACM Int. Conf. Multimedia*, 2024, pp. 3657–3665.

[16] X. Lou, S. Guo, J. Li, and T. Zhang, "Ownership verification of DNN architectures via hardware cache side channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 8078–8093, Nov.2022.

[17] H. Luo, L. Li, and J. Li, "Digital watermarking technology for AI-generated images: A survey," *Mathematics*, vol. 13, no. 4, p. 651, Feb.2025.

[18] M. Mo, C. Wang, Q. Guo, S. Bian, Q. Huang, and X. Zhang, "A novel robust black-box fingerprinting scheme for deep classification neural networks," *Expert Syst. Appl.*, vol. 252, Oct.2024, Art. no. 124201.

[19] H. Nie and S. Lu, "FedCRMW: Federated model ownership verification with compression-resistant model watermarking," *Expert Syst. Appl.*, vol. 249, Sep.2024, Art. no. 123776.

[20] K. Arabi, B. Feuer, R. Teal Witter, C. Hegde, and N. Cohen, "Hidden in the noise: Two-stage robust watermarking for images," 2024, *arXiv:2412.04653*.

[21] H. Ci, P. Yang, Y. Song, and M. Z. Shou, "RingID: Rethinking tree-ring watermarking for enhanced multi-key identification," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 338–354.

[22] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-rings watermarks: Invisible fingerprints for diffusion images," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 58047–58063.

[23] Z. Yang, K. Zeng, K. Chen, H. Fang, W. Zhang, and N. Yu, "Gaussian shading: Provable performance-lossless image watermarking for diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 12162–12171.

[24] H. Ci, Y. Song, P. Yang, J. Xie, and M. Zheng Shou, "WMAdapter: Adding watermark control to latent diffusion models," 2024, *arXiv:2406.08337*.

[25] D. Lin, Y. Li, B. Tondi, K. Lin, B. Li, and M. Barni, "An efficient watermarking method for latent diffusion models via low-rank adaptation and dynamic loss weighting," 2024, *arXiv:2410.20202*.

[26] C. Kim, K. Min, M. Patel, S. Cheng, and Y. Yang, "WOUAF: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 8974–8983.

[27] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 22409–22420.

[28] C. Xiong, C. Qin, G. Feng, and X. Zhang, "Flexible and secure watermarking for latent diffusion model," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 1668–1676.

[29] W. Feng et al., "AquaLoRA: Toward white-box protection for customized stable diffusion models via watermark LoRa," 2024, *arXiv:2405.11135*.

[30] R. Min, S. Li, H. Chen, and M. Cheng, "A watermark-conditioned diffusion model for IP protection," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 104–120.

[31] Y. Liu, Z. Li, M. Backes, Y. Shen, and Y. Zhang, "Watermarking diffusion model," 2023, *arXiv:2305.12502*.

[32] Z. Yuan, L. Li, Z. Wang, and X. Zhang, "Watermarking for stable diffusion models," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 35238–35249, Nov.2024.

[33] Y. Zhao, T. Pang, C. Du, X. Yang, N.-M. Cheung, and M. Lin, "A recipe for watermarking diffusion models," 2023, *arXiv:2303.10137*.

[34] Z. Yuan, L. Li, Z. Wang, and X. Zhang, "Ambiguity attack against text-to-image diffusion model watermarking," *Signal Process.*, vol. 221, Aug.2024, Art. no. 109509.

[35] J. E. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2021, p. 3.

[36] V. Dimitri, B. Regina, and M. Alfonz, "A survey on mixture of experts: Advancements, challenges, and future directions," *TechRxiv*, Jan. 2025.

[37] Y. Adi, C. Baum, M. Cissé, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 1615–1631.

[38] Y. Quan, H. Teng, Y. Chen, and H. Ji, "Watermarking deep neural networks in image processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1852–1865, May2021.

[39] J. Zhang et al., "Protecting intellectual property of deep neural networks with watermarking," in *Proc. Asia Conf. Comput. Commun. Secur.*, 2018, pp. 159–172.

[40] S. Peng, Y. Chen, C. Wang, and X. Jia, "Intellectual property protection of diffusion models via the watermark diffusion process," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2023, pp. 290–305.

[41] Q. Li and E. Chang, "Zero-knowledge watermark detection resistant to ambiguity attacks," in *Proc. 8th workshop Multimedia Secur.*, vol. 2939, 2006, pp. 158–163.

[42] K. Loukhaoukha, A. Refaey, and K. Zebbiche, "Ambiguity attacks on robust blind image watermarking scheme based on redundant discrete wavelet transform and singular value decomposition," *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 3, pp. 359–368, Dec.2017.

[43] Z. Yuan, L. Li, Z. Wang, and X. Zhang, "Protecting copyright of stable diffusion models from ambiguity attacks," *Signal Process.*, vol. 227, Feb.2025, Art. no. 109722.

[44] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[45] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7120–7129.

[46] N. Du et al., "GLaM: Efficient scaling of language models with mixture-of-experts," in *Proc. Int. Conf. Mach. Learn.*, vol. 162, 2022, pp. 5547–5569.

[47] F. Xue et al., "OpenMoE: An early effort on open mixture-of-experts language models," 2024, *arXiv:2402.01739*.

[48] H. Hazimeh et al., "DSelect-k: Differentiable selection in the mixture of experts with applications to multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 29335–29347.

[49] D. Lepikhin et al., "GShard: Scaling giant models with conditional computation and automatic sharding," 2020, *arXiv:2006.16668*.

[50] K. R. N. Reddy and S. F. Basha, "Real time object identification: A study on COCO dataset," in *Proc. Int. Conf. Adv. Mater.*, 2025, pp. 860–870.

[51] J. N. M. Pinkney. (2022). *Pokemon BLIP Captions*. [Online]. Available: https://huggingface.co/datasets/reach-vb/pokemon-blip-captions

[52] S. N. Ramesh and Z. Zhao, "Efficient pruning of text-to-image models: Insights from pruning stable diffusion," 2024, *arXiv:2411.15113*.