A Collusion Attack on Stable Signature and a Defense using Domain-based Signature Assignment

Invited Paper

Spandana Korabandi*, Alimu Alibotaiken*, Suyang Wang†, Yu Cheng*
*Department of Electrical and Computer Engineering,
Illinois Institute of Technology, Chicago, IL 60616, USA
†School of Computer Science and Engineering,
California State University, San Bernardino, San Bernardino, CA 92407, USA
Email: {skorabandi, aalibotaiken}@hawk.illinoistech.edu, suyang.wang@csusb.edu, cheng@illinoistech.edu

***Abstract*—Stable Signature is a recent watermarking framework based on latent diffusion models, which generates images with embedded signatures by fine-tuning the decoder. While prior work has shown that watermarks can be removed while maintaining visual quality by retraining the watermarked decoder with clean images, we demonstrate that collusion among multiple users poses a practical and severe threat. Our attack begins by averaging watermarked decoders, which already provides a strong initialization for watermark removal. With encoder access, this initialization can be further fine-tuned to significantly suppress the watermark signal. Even when the encoder is not available, colluders can expand their group size to achieve comparable effectiveness, highlighting the scalability of the attack. To defend against this threat, we propose a domain-based signature assignment mechanism. In this strategy, the watermarking service provider (e.g., one using Stable Signature) partitions the signature space into domains, requiring all users in the same domain to share a fixed set of domain-index bits in their signatures. Experiments show that the domain-index bits remain robust under the collusion attack when the encoder is not available. Our studies suggest that adopting the domain-based signature assignment and keeping the encoder confidential will be good practices when Stable Signature is used as a watermarking solution.**

***Index Terms*—Image watermarking, Stable Signature, collusion attack, domain-based signature assignment.**

I. Introduction

The rapid growth of generative AI [1]–[4] has intensified concerns about authenticity and attribution of AI-generated content. Watermarking methods address this by embedding identifiable signals into generated outputs. Among them, Stable Signature [5] embeds user-specific binary keys directly into a latent diffusion model's decoder, enabling per-user attribution without relying on the encoder or inference-time perturbations. It has demonstrated robustness against common image-level transformations such as cropping and compression.

However, most existing attack studies focus on image-level manipulation or single-user model-level purification [6]–[9]. The recent work by Hu et al. [10] shows that fine-tuning a watermarked decoder on clean images can effectively suppress the embedded key, but this approach is computationally expensive, especially in encoder-agnostic settings where latent optimization must be performed per image. Meanwhile, Stable Signature briefly mentions the possibility of decoder collusion but leaves the problem largely unexplored.

In this paper, we present the first systematic study of multi-user collusion attacks on Stable Signature. We show that averaging multiple uniquely watermarked decoders already weakens the watermark signal by canceling independent perturbations. We further introduce a lightweight fine-tuning procedure that uses only watermarked images generated by the colluding models themselves, allowing effective watermark removal under both encoder-aware and encoder-agnostic scenarios without requiring clean datasets. To defend against such attacks, we propose a *domain-based signature assignment* strategy that embeds shared prefix bits across users. This introduces correlated perturbations that resist cancellation during averaging and significantly improves robustness, particularly when attackers lack access to the encoder. The source code for this work is available online[1].

Our main contributions are:

1) We propose and systematically study, for the first time, a model collusion attack involving multiple users on Stable Signature framework, showing that model averaging followed by proper fine-tuning can effectively remove watermarks in both encoder-aware and encoder-agnostic settings.
2) We demonstrate that this collusion attack can be lightweight, as with an appropriately chosen loss function, fine-tuning requires only the colluders' watermarked images as training data, thereby avoiding the need for clean data or costly latent optimization.
3) We conduct a detailed empirical evaluation of Stable Signature framework under multi-user collusion, revealing that even a small number of colluding users can substantially weaken the watermark signal.
4) We propose a simple yet effective defense, **Domain-based Signature Assignment**, which introduces shared bits across watermark keys to prevent complete perturbation cancellation during averaging.
5) We show that domain-based signature assignment improves robustness against collusion, particularly under

[1]https://github.com/FUNLAB-IIT/stable-signature-collusion-and-defense

979-8-3315-7111-5/26/$31.00 ©2026 IEEE

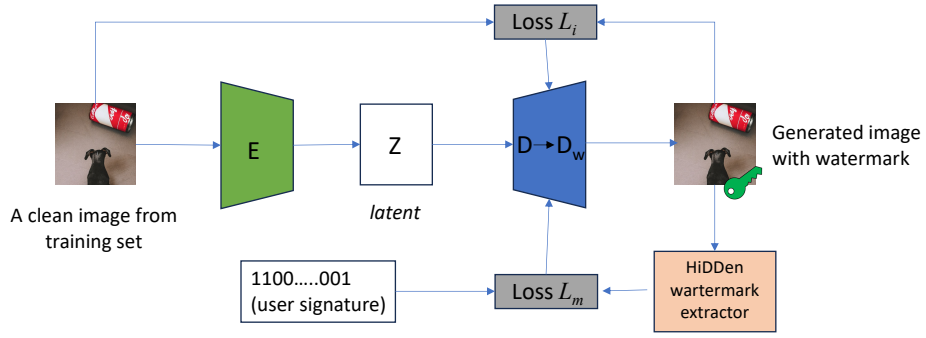

Fig. 1. Overview of the Stable Signature's Framework: fine-tuning the latent Decoder D to the watermarked decoder D_w.

encoder-agnostic conditions, with minimal changes to the existing watermarking pipeline.

This paper is organized as follows. Section II provides background and related work on latent diffusion models and Stable Signature. Section III defines the threat model, including attacker capabilities and goals. Section IV presents our proposed collusion attack method, covering model averaging, decoder fine-tuning, and data collection strategies. Section V introduces our domain-based signature assignment defense. Section VI reports experimental results evaluating the effectiveness of the attack and the robustness of the defense. Finally, Section VII concludes the paper.

II. Background

Latent diffusion models (LDMs) [11] generate images by denoising latent variables produced by a pre-trained autoencoder. Since the decoder directly reconstructs the final image from the latent representation, it becomes a natural location for embedding watermark information.

Stable Signature fine-tunes the LDM decoder to embed a user-specific 48-bit binary message. The method adopts the HiDDen framework [12], where the decoder is optimized using a combination of message reconstruction loss and perceptual loss:

$$L = L_m + \lambda_i L_i, \tag{1}$$

ensuring that the embedded message is recoverable while maintaining high perceptual fidelity. Because the watermark is baked into the decoder weights, no encoder access or inference-time perturbation is required.

A trained extractor network recovers the message from generated images. Following the design in Stable Signature, a key is considered correctly detected if at least 41 out of the 48 bits match, corresponding to an 86% bitwise accuracy threshold. This setting yields a false positive rate below 10^{-6}, making it highly unlikely for an unmarked or incorrectly marked image to be mistaken as valid.

Research on defeating watermarks generally falls into two areas. Image-level attacks alter the final output through transformations or adversarial removal [6]–[9]. Model-level attacks modify the generator itself, such as through decoder purification or latent-space fine-tuning [10]. Work on Stable Signature has also considered network-level threats like purification and pairwise collusion, but only in limited settings without large-scale or systematic collusion.

Most prior methods assume a single-user threat model, attacking instances independently. This leaves a key gap: coordinated adversaries controlling multiple uniquely watermarked decoders. We address this by studying collusion attacks, showing they can reliably suppress watermark signals while preserving image quality. While Hu et al. [10] show that optimization-based fine-tuning can strongly remove watermarks, it demands clean data and heavy computation, especially in encoder-agnostic scenarios where optimization must be repeated per image. In contrast, collusion is lightweight, using only watermarked models and shared colluders' images, making it a practical and scalable threat that prior work has largely overlooked.

III. Threat Model

We consider a scenario in which a total of N users participate in the collusion as shown in Fig 2, each possessing a uniquely fine-tuned watermarked decoder $D_w^{(i)}$ that has been fine-tuned from a base decoder D to embed a distinct 48-bit binary watermark key, where $i = 1, 2, \ldots, N$. The watermark is embedded directly into the decoder's parameters, enabling in-generation watermarking as proposed by Stable Signature. Note that our threat model follows a structure that is similar to that proposed in [10], analyzing encoder-aware and encoder-agnostic attacks on watermarked diffusion decoders. However, instead of focusing on watermark removal via image supervision training, we propose a collusion attack leveraging multiple uniquely watermarked decoders.

A. Attacker's Goal

The goal of a colluding group of users is to remove the watermark from their decoders $D_w^{(i)}$. Specifically, the attackers aim to construct a new decoder D_{attack} that can not only generate non-watermark images but also preserve the high perceptual quality of the generated images.

B. Attacker's Knowledge

We assume the attacker has access to their own uniquely watermarked decoders $D_w^{(i)}$ for $i \in \{1, 2, \ldots, N\}$, a collection of independently generated watermarked images that serve as a training dataset, and the extractor model used to decode watermark bits from generated images.

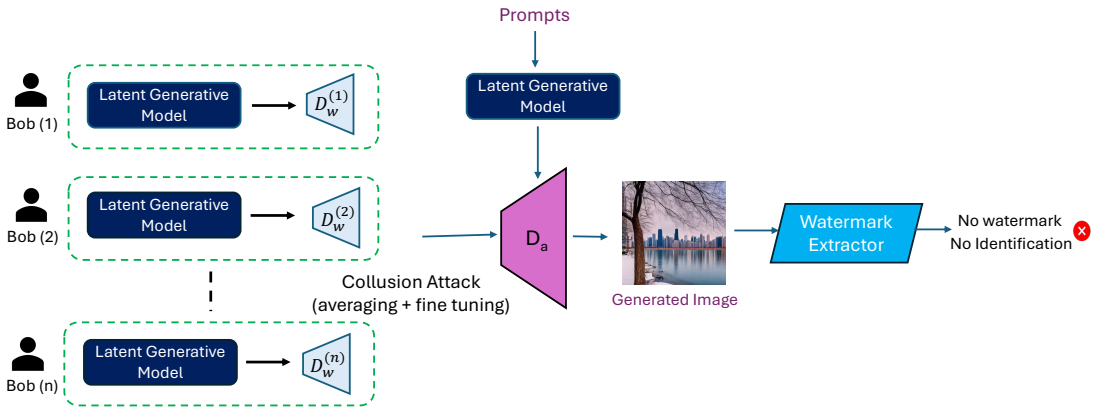

Fig. 2. Overview of the collusion attack: Multiple users, each processing a uniquely watermarked decoder, collaborate to generate a new compromised decoder D_a. This colluded decoder D_a is then used in the latent generative model pipeline to produce high-quality images that bypass the watermark extractor, resulting in failed detection and identification.

Depending on the attacker's access to upstream components of the latent diffusion pipeline, we consider two scenarios. In the **encoder-agnostic** scenario, the attacker lacks access to the encoder and denoising modules but does have the latent vectors z used for generation (e.g., captured during the forward pass of watermarked image generation). In the **encoder-aware** scenario the attacker has access to the encoder and diffusion process, enabling fast and scalable latent extraction for arbitrary images and allowing the attacker to use both watermarked images produced by $D_w^{(i)}$ and clean public datasets (e.g., ImageNet) for decoder fine-tuning, which facilitates more flexible and large-scale attacks.

C. Attacker's Capability

We assume that the attackers are able to modify the parameters of their own decoder weights $D_w^{(i)}$.

In summary, this threat model reflects a realistic and potentially damaging scenario: a small number of colluding users, even with limited knowledge of the full pipeline, may still successfully remove the watermark while preserving image quality.

IV. Collusion Attack Method

Based on the threat model introduced in Section III, we now present our collusion watermark removal strategy. In this setting, each colluding user holds a uniquely watermarked decoder $D_w^{(i)}$, whose weights we denote as $W^{(i)} = W + \Delta^{(i)}$, where W represents the parameters of the clean (non-watermarked) decoder, and $\Delta^{(i)}$ is the perturbation introduced to embed the watermark key. The group of attackers aims to construct a new decoder that suppresses watermark signals while maintaining image quality.

Let $W \in \mathbb{R}^d$ denote the parameter vector of the clean decoder. The watermarked decoder for the i-th user is given by:

$$W^{(i)} = W + \Delta^{(i)}, \tag{2}$$

where $\Delta^{(i)} \in \mathbb{R}^d$ is the watermark-specific perturbation, and d is the total number of decoder parameters. The addition is performed element-wise.

Since the 48-bit watermark keys are randomly sampled per user and the fine-tuning process preserves generative quality, the perturbations $\Delta^{(i)}$ are expected to be small in magnitude and symmetrically distributed, and relatively minor in scale.

In practice, the outputs of different $D_w^{(i)}$ appear perceptually similar, as shown in Fig. 3, reflecting that $\Delta^{(i)}$ does not significantly shift the model's image distribution.

Now consider a scenario where N users collude by averaging their watermarked model weights. The resulting decoder has weights:

$$W_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} W^{(i)} = W + \frac{1}{N}\sum_{i=1}^{N} \Delta^{(i)}. \tag{3}$$

If the perturbations $\Delta^{(i)}$ are approximately zero-mean and uncorrelated across users, the aggregated perturbation term tends to cancel out as N increases:

$$W_{\text{avg}} \approx W. \tag{4}$$

This implies that averaging multiple watermarked decoders can reduce or eliminate the embedded watermark signal, since user-specific perturbations destructively interfere under ideal statistical assumptions. As shown in Fig. 4, collusion already lowers the bit accuracy to 0.81, much lower than other attack baselines. This substantial drop, however, still leaves a detectable watermark, which motivates us to go beyond averaging.

To address the limitations of averaging alone, we propose a two-stage attack method under both encoder-aware and encoder-agnostic settings. In the first stage (Averaging), we compute W_{avg} as the average of the weights from N watermarked decoders, providing an initialization that weakens the watermark signal by partially canceling diverse perturbations. In the second stage (Fine-tuning), the decoder initialized with W_{avg} is further optimized using a dataset of watermarked images generated by $D_w^{(i)}$ (detailed in the following subsection), which helps remove residual watermark signals while preserving image quality. This strategy leverages the fact that averaging alone can suppress much of the watermark signal

but not always completely eliminate it, whereas fine-tuning completes the removal by explicitly optimizing the model in the image space.

Fig. 3. Images generated in order (from left to right) by Decoder with W, $D_w^{(1)}$, $D_w^{(2)}$, Decoder with W_{avg}.

A. Attacker Dataset Setup

To fine-tune the decoder after getting D_{avg} according to equation (3), we next construct a dataset by $D_w^{(i)}$ following Stable Signature's pipeline. This approach allows attackers to generate training data without relying on original or external datasets.

Each data in the dataset consists of a latent vector z_j and its corresponding watermarked image $x_{w,j}$, such that

$$x_{w,j}^{(i)} = D_w^{(i)}(z_j), \quad \forall i \in \{1, \ldots, N\}, \forall j \in \{1, \ldots, M_i\}. \quad (5)$$

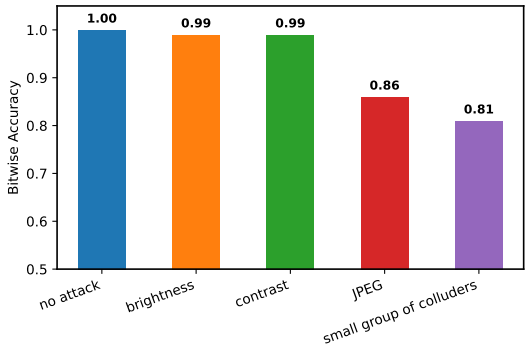

Fig. 4. Bitwise accuracy across different attacks on the random prompts generated dataset.

As shown in equation (5), $(z_j, x_{w,j}^{(i)})$ denotes the j-th latent–image pair generated using $D_w^{(i)}$, where $j \in \{1, \ldots, M_i\}$ and M_i is the number of samples generated by user i. These latent vectors z_j are captured using diverse prompts and noise seeds to ensure broad coverage of the latent space.

As shown in Figure 5, latent vectors z_j are generated and passed through different watermarked decoders $D_w^{(i)}$ to produce the corresponding images $x_{w,j}^i$. The resulting $(z_j, x_{w,j}^{(i)})$ pairs are collected and used to supervise decoder fine-tuning.

Note that the $(z_j, x_{w,j}^{(i)})$ pairs are particularly for the encoder-agnostic setting, since there is no encoder for latent mapping. For the encoder-aware setting, we use only watermarked images $x_w^{(i)}$ as our dataset because the attackers have latent mapping ability with the encoder.

B. Post-collusion Fine-tuning

The final step of our attack refines the averaged decoder D_{avg} obtained from collusion by fine-tuning it using the

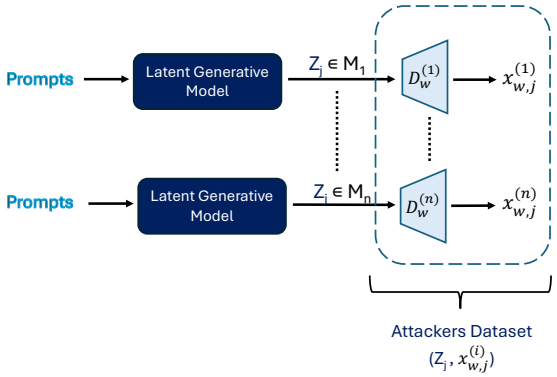

Fig. 5. Generation of attacker data: the same prompts are processed through different decoders with unique signatures, resulting in images with distinct watermark signatures.

attackers' dataset we collected as described in previous subsection IV (A). The goal is to further remove the residual watermark signals while preserving image quality.

We denote the fine-tuned decoder as D_a. The fine-tuning process optimizes D_a using gradient descent on a reconstruction loss between the decoder's output and the target watermarked image. Specifically, for each latent–image pair $(z_j, x_{w,j}^{(i)})$, the decoder is trained to minimize the following objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda \cdot \mathcal{L}_{\text{perc}}, \quad (6)$$

where $\mathcal{L}_{\text{rec}}$ is a pixel-wise loss (e.g., MSE), $\mathcal{L}_{\text{perc}}$ is a perceptual loss, and λ controls the weight of perceptual alignment. We use perceptual loss by default, but our framework supports other choices such as SSIM, LPIPS, or a combination of these.

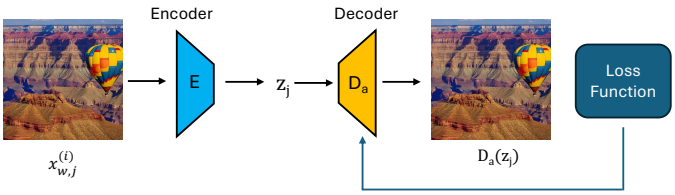

Fig. 6. Overview of the encoder-aware scenario: Encoder E is available to the attacker, each image $x_{w,j}^{(i)}$ in the attacker's dataset passes through the encoder to provide z_j. This latent is passed through the attack decoder. The weights of the attack decoder are updated based on the loss function.

This fine-tuning is performed under two scenarios:

a) Encoder-aware: In this case, as illustrated in Fig. 6, the attacker has access to the encoder of the latent diffusion model. Instead of storing precomputed latent vectors, the attacker only retains the watermarked images $x_w^{(i)}$ and uses the encoder to extract their corresponding latent vectors on-the-fly. This ensures that the latent vectors are consistent with the model's encoding process, enabling more precise training. The attack decoder D_a is trained using these freshly encoded $(E(x_w^{(i)}), x_w^{(i)})$ pairs. The detail is shown in algorithm 1.

b) Encoder-agnostic: In this setting, as illustrated in Fig. 7, the attacker lacks access to the encoder. As a result, the attacker will directly use latent–image pairs $(z_j, x_{w,j}^{(i)})$

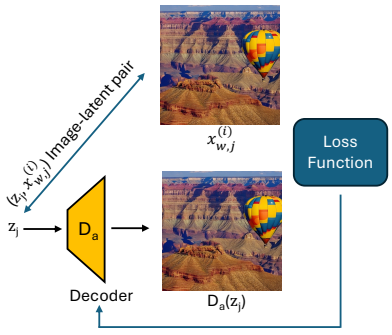

Fig. 7. Overview of the encoder-agnostic Scenario: Encoder E is not available to the attacker, the saved latent-image pair $(z_j, x_{w,j}^{(i)})$. This latent is passed through the attack decoder. The weights of the attack decoder are updated based on the loss function.

previously generated and saved during the dataset construction. This approach avoids the cost of re-encoding, making it efficient and practical for resource-constrained adversaries. The detail is shown in algorithm 2.

Algorithm 1 Encoder-aware

Require: Images $x_w^{(i)}$ from attackers' dataset, Decoder weights $W^{(1)}, W^{(2)}, \ldots, W^{(n)}$

1: Compute averaged decoder weights: $W_{\text{avg}} \leftarrow \frac{1}{n}\sum_{i=1}^{n} W^{(i)}$
2: Initialize attack decoder weights: $W_a \leftarrow W_{\text{avg}}$
3: Compute loss according to equation 6
4: Update weights: $W_a \leftarrow W_a - \eta \cdot \nabla_{W_a}\mathcal{L}$
5: Repeat Steps 3–4 until convergence
6: Save final decoder D_a weights W_a

Algorithm 2 Encoder-agnostic

Require: Latent-Image pairs $(E(x_w^{(i)}), x_w^{(i)})$, Decoder weights $W^{(1)}, W^{(2)}, \ldots, W^{(n)}$

1: Compute averaged decoder weights: $W_{\text{avg}} \leftarrow \frac{1}{n}\sum_{i=1}^{n} W^{(i)}$
2: Initialize attack decoder weights: $W_a \leftarrow W_{\text{avg}}$
3: Compute loss according to equation 6
4: Update weights: $W_a \leftarrow W_a - \eta \cdot \nabla_{W_a}\mathcal{L}$
5: Repeat Steps 3–4 until convergence
6: Save final decoder D_a weights W_a

V. Defense via Domain Index Assignment

To mitigate strong collusion attacks, we propose a key assignment strategy called *Domain Index Assignment*. This method introduces correlation across watermark keys by sharing partial key bits among users, thereby breaking the assumption that $\Delta^{(i)}$ are independent and zero-mean. This assumption is critical to the success of averaging-based attacks. This approach is similar to the idea of IP address allocation within a network: the high-order bits (network prefix) are fixed to define the subnet, while the low-order bits (host ID) are varied per user. In our context, domain-based signature assignment ensures that part of the watermark key is consistent across users. This bit-sharing prevents cancellation through averaging.

Each user-specific decoder $D_w^{(i)}$ is embedded with a 48-bit key $k^{(i)} \in \{0,1\}^{48}$. Instead of sampling all 48 bits independently, the key is split into

$$k^{(i)} = \left[k_{\text{fixed}},\ k_{\text{rand}}^{(i)}\right], \tag{7}$$

where $k_{\text{fixed}} \in \{0,1\}^n$ is a secret prefix shared by all users and $k_{\text{rand}}^{(i)} \in \{0,1\}^{48-n}$ is user-specific. This reduces the key space to 2^{48-n}, balancing robustness and uniqueness: larger n yields stronger correlation (robustness), smaller n more entropy.

Without defense, keys are sampled uniformly, giving perturbations $\Delta^{(i)} \sim \mathcal{U}(-\alpha,\alpha)^d$, so that $\frac{1}{N}\sum_{i=1}^{N}\Delta^{(i)} \approx 0$ and averaging cancels the watermark. With domain-based assignment, perturbations are split as

$$\Delta^{(i)} = \Delta_{\text{shared}} + \Delta_{\text{unique}}^{(i)}, \tag{8}$$

where Δ_{shared} is fixed across users. The average becomes

$$\frac{1}{N}\sum_{i=1}^{N}\Delta^{(i)} = \Delta_{\text{shared}} + \frac{1}{N}\sum_{i=1}^{N}\Delta_{\text{unique}}^{(i)}, \tag{9}$$

so the first term preserves watermark bias even as the second term vanishes. Thus, averaging no longer cancels the signal, invalidating the core assumption behind collusion attacks.

VI. Experimental Results

Dataset: We used 3000 image–latent pairs generated from 1000 prompts, each rendered with three different decoders to simulate randomly watermarked outputs. Source images were 512×512; for encoder-aware training they were downsampled to 256×256, while encoder-agnostic training used the original 512×512 images with 64×64 latents. Watermarked decoders were created by fine-tuning the clean Stable Diffusion-2 decoder on 2000 MS-COCO images, yielding 10 decoders each with a unique 48-bit watermark. A disjoint test set of 100 images per method was generated from non-overlapping prompts.

Parameter Settings: Training and evaluation were performed on an NVIDIA RTX 4070 Ti GPU. We followed Stable Signature hyperparameters (architecture, learning rate, batch size, optimizer) and employed MSE as reconstruction loss with Watson-VGG as perceptual loss. In our experiments, we define a small colluder group as 3 users and a large colluder group as 10 users.

Baselines: We compared against three per-image removal techniques such as contrast, JPEG compression, and brightness adjustment.

Metrics: Performance was measured by bitwise accuracy (fraction of extracted bits matching the ground-truth key) and image quality using FID, SSIM, and PSNR. For each test image, outputs were compared to the corresponding watermarked Stable Diffusion-2 image generated with the same seed [10]. Final results report averages over the 100 test images.

TABLE I
COMPARISON WITH OTHER ATTACKS

Attack Method	FID ↓	PSNR ↑	SSIM ↑	Bit accu ↓
Brightness	185.45	5.07	0.36	0.99
Contrast	42.10	16.61	0.64	0.99
JPEG compression	29.10	28.56	0.83	0.86
W_{avg} of small group of colluders	10.95	30.24	**0.9**	0.75
W_{avg} of large group of colluders	**6.27**	**30.57**	**0.9**	0.65
Model purification (MP)	18.66	26.99	0.80	0.89
Encoder-aware under small group of colluders	19.30	26.61	0.79	**0.61**
Encoder-agnostic under small group of colluders	14.68	27.97	0.85	0.74

TABLE II
COMPARISON OF ATTACK PERFORMANCE WITH DOMAIN-BASED INDEX ASSIGNMENT

Method	FID ↓	PSNR ↑	SSIM ↑	Bit Acc ↓	Last 10-Bit Acc ↓
W_{avg} of large group of colluders	4.46	31.70	0.92	0.92	0.64
Encoder-aware	17.92	26.84	0.79	0.64	0.67
Encoder-agnostic	20.49	28.36	0.89	0.93	0.68

A. Collusion Effectiveness

We first evaluate collusion attacks under a small group of colluders. As shown in Table I, both encoder-aware and encoder-agnostic variants significantly weaken the watermark while preserving high image quality, and Fig. 8 shows that the generated images remain visually close to Stable Diffusion 2 outputs. With encoder access, fine-tuning poses a severe threat, driving bit accuracy down from 1.00 to 0.61 with SSIM 0.79 and FID 19.30. The encoder-agnostic case achieves smaller gains from fine-tuning, yet its danger should not be underestimated: by simply expanding the colluder group, accuracy drops even further to 0.65, exposing a critical vulnerability in watermark robustness.

We further investigate whether increasing the dataset size of the colluding users improves watermark removal. Figure 9 shows bitwise accuracy across datasets of different sizes (1000, 3000, and 6000 images) under both encoder-aware and encoder-agnostic settings. The results indicate that extending the colluders' dataset size does not significantly improve performance: the curves remain nearly flat across different sizes.

This observation suggests that dataset scaling provides limited benefit to colluders, and that the key factor for effectiveness lies instead in increasing the number of colluders.

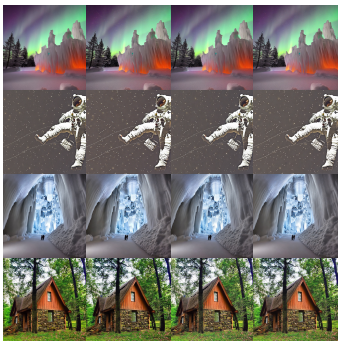

Fig. 8. Comparison of different attacks: images generated in the order (left to right) by Stable Diffusion 2 model, Stable Signature watermarked decoder ($D_w^{(1)}$), our encoder-aware attack decoder, and our encoder-agnostic attack decoder.

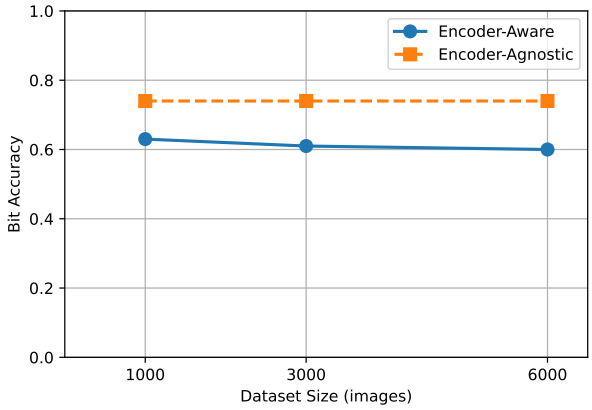

Fig. 9. Bitwise accuracy of encoder-aware and encoder-agnostic collusion attacks under varying dataset sizes (1000, 3000, and 6000 images). Results show that enlarging the dataset has little effect on performance, confirming that expanding the number of colluders is more effective than scaling individual datasets.

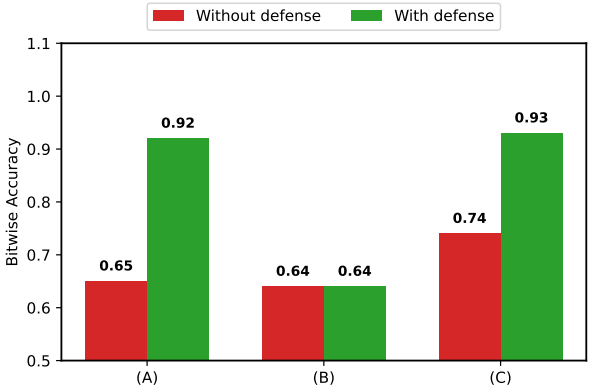

Fig. 10. Performance of domain-based signature assignment: (A) is the comparison between W_{avg} of a large group of colluders, (B) is the comparison under encoder-aware, and (C) is the comparison under encoder-agnostic.

B. Domain-based Assignment Defense Effectiveness

To evaluate our defense, we tested watermarked models under collusion using domain-based signature assignment with 38 fixed bits and 10 random bits (Table II, Fig. 10). The results show a clear improvement: bit accuracy remains as high as 0.93 in the encoder-agnostic case and 0.92 for the large group of colluders' averaged decoder D_{avg} with fixed bits, compared to only 0.65 without defense. Even when attackers target the last 10 bits, the full 48-bit key persists, demonstrating strong protection against collusion while preserving image quality.

Our defense is especially effective against large-scale collusion in the encoder-agnostic scenario, where attackers are limited to latent–image pairs. In contrast, the encoder-aware setting is less robust, since access to the encoder enables attackers to generate diverse training pairs that strengthen fine-tuning by using clean images. This gap highlights an important takeaway for model providers: restricting access to the encoder is critical for maintaining watermark robustness.

Compared to prior work (Table I), our method consistently resists collusion attacks while sustaining high perceptual

TABLE III
ACCURACY OF FIXED BITS, NUMBER OF k_{FIXED} IS 38-BIT, AND REMAINING BITS UNDER LARGE-GROUP ($N = 10$) COLLUSION. THE FIXED-BIT ACCURACY REMAINS HIGH FOR THE CORRECT DOMAIN, ENABLING GROUP IDENTIFICATION.

k_{fixed} (Domain)	Fixed-bit Acc.	Remaining-bit Acc.
Domain 1	0.97	0.42
Domain 2	0.98	0.50
Domain 3	0.97	0.57

quality. By embedding shared bits across users, domain-based signature assignment blocks the perturbation cancellation that underlies averaging attacks.

Besides improving robustness, domain-based signature assignment enables the defender to determine *which group the colluders belong to*. Each domain is defined by a distinct k_{fixed}: *Domain 1* uses a simple pattern where the first half of k_{fixed} bits are all 0's and the remaining half are all 1's; *Domain 2* uses a more structured pattern with long runs of 1's and a segment of periodically alternating 0's and 1's; and *Domain 3* uses a fully random k_{fixed} pattern. To evaluate domain recognition, we treat an extraction as successful only when the recovered k_{fixed} achieves at least 90% accuracy, which is an intentionally high standard. We then assess success rates over 100 images.

After a strong $N = 10$ collusion attack, as shown in Table III, accuracy on the true domain's k_{fixed} remains high (approximately 0.97–0.98), while all other domains drop to near-random levels (approximately 0.42–0.57). This separation yields consistent success under the 90% criterion, allowing reliable identification of the colluders' domain even when user-specific bits are largely destroyed.

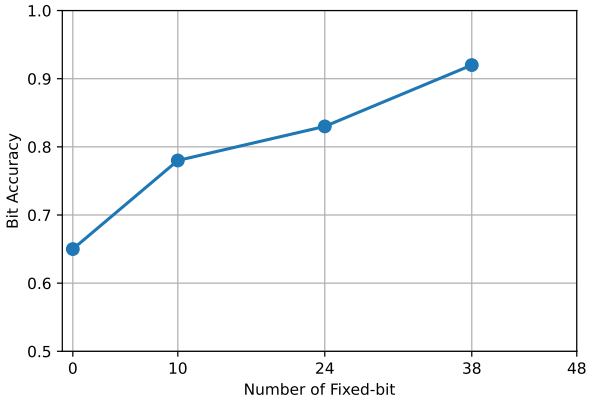

Fig. 11. Bit accuracy under collusion attack with different numbers of fixed bits. More fixed bits result in stronger robustness against large group collusion averaging.

To further study the robustness of domain-based signature assignment, we vary the number of k_{fixed} from 0 (no defense) to 38 (the strongest setting used in the main paper). Figure 11 shows the bit accuracy under collusion attacks as the number of k_{fixed} increases. The trend reveals that even a small number of k_{fixed} (e.g., 10) already improves robustness substantially compared to large group collusion, and the protection continues to strengthen as more bits are k_{fixed}. This supports our intuition: increasing correlation across users' signatures preserves a stronger shared watermark signal that resists cancellation during averaging.

VII. CONCLUSION

We analyzed collusion attacks on Stable Signature and showed that averaging and fine-tuning watermarked decoders can remove embedded signatures, with larger colluder groups strengthening encoder-agnostic attacks. To mitigate this, we proposed domain-based signature assignment, which embeds shared bits across users to retain robustness. Our results underline the practicality of collusion attacks and the need for restricted encoder access and domain-based defenses.

ACKNOWLEDGMENT

This work was supported in part by the NSF under grant CNS-2008092.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, Oct. 2020. [Online]. Available: https://doi.org/10.1145/3422622

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[3] C. Saharia, W. Chan, S. Saxena, L. Lit, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. Gontijo-Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022.

[4] P.-C. Chen, O. T. Ajayi, and Y. Cheng, "Large language model based machine learning approach for fake news detection: Invited paper," in *2024 International Conference on Meta Computing (ICMC)*, 2024, pp. 241–249.

[5] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 466–22 477.

[6] X. Zhao, K. Zhang, Z. Su, S. Vasan, I. Grishchenko, C. Kruegel, G. Vigna, Y.-X. Wang, and L. Li, "Invisible image watermarks are provably removable using generative AI," *Advances in neural information processing systems*, vol. 37, pp. 8643–8672, 2024.

[7] Z. Jiang, J. Zhang, and N. Z. Gong, "Evading watermark based detection of ai-generated content," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1168–1181.

[8] M. Saberi, V. S. Sadasivan, K. Rezaei, A. Kumar, A. Chegini, W. Wang, and S. Feizi, "Robustness of ai-image detectors: Fundamental limits and practical attacks," *arXiv preprint arXiv:2310.00076*, 2023.

[9] N. Lukas, A. Diaa, L. Fenaux, and F. Kerschbaum, "Leveraging optimization for adaptive attacks on image watermarks," *arXiv preprint arXiv:2309.16952*, 2023.

[10] Y. Hu, Z. Jiang, M. Guo, and N. Gong, "Stable signature is unstable: Removing image watermark from diffusion models," *arXiv preprint arXiv:2405.07145*, 2024.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[12] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Computer Vision – ECCV 2018*, ser. Lecture Notes in Computer Science, vol. 11219. Cham, Switzerland: Springer, 2018, pp. 682–697. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-01267-0_40