This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Dynamic watermarks in images generated by diffusion models

Yunzhuo Chen[1],Jordan Vice[1], Naveed Akhtar[2,1], Nur Al Hasan Haldar[3,1], Ajmal Mian[1]
[1]The University of Western Australia, Perth, Australia
[2]The University of Melbourne, Melbourne, Australia
[3]Curtin University, Perth, Australia
yunzhuo.chen@research.uwa.edu.au, naveed.akhtar1@unimelb.edu.au,
nur.haldar@curtin.edu.au , jordan.vice@uwa.edu.au, ajmal.mian@uwa.edu.au

Abstract

High-fidelity text-to-image diffusion models have revolutionized visual content generation, but their widespread use raises significant copyright concerns. We propose a novel multi-stage watermarking framework for diffusion models, designed to establish copyright and trace generated images back to their source. Our technique involves embedding: (i) a fixed watermark that is localized in the diffusion model's learned noise distribution and, (ii) a human-imperceptible, dynamic watermark in generates images, leveraging a fine-tuned decoder. By leveraging the Structural Similarity Index Measure (SSIM) and cosine similarity, we adapt the watermark's shape and color to the generated content while maintaining robustness. We demonstrate that our method enables reliable source verification through watermark classification, even when the dynamic watermark is adjusted for content-specific variations. Source model verification is enabled through watermark classification. We generate a dataset of watermarked images and introduce a methodology to evaluate the statistical impact of watermarking on generated content. Additionally, we rigorously test our framework against various attack scenarios, demonstrating its robustness and minimal impact on image quality.

1. Introduction

Recent deep generative models have made photorealistic image generation more accessible [2, 22, 31, 38], as exemplified by DALL·E 2 [29], Stable Diffusion [30], and FLUX models [24]. These models have also spurred the development of numerous image editing tools and text-to-video models, including ControlNet [37], Instruct-Pix2Pix [3], and SORA [4]. While these models represent valuable digital assets, their widespread use raises the potential for misuse in spreading false narratives, generating harmful representations, or infringing intellectual property (IP) rights.

To address these challenges, blind watermarking has emerged as a key strategy for IP protection and misuse prevention. However, current watermarking methods face limitations, as fixed watermarks embedded during training are often predictable and vulnerable to removal or tampering through reverse engineering or image-processing techniques [2, 22, 27, 31, 38]. We propose a novel dual watermarking strategy by embedding: (*i*) a unique, model-specific QR-code watermark directly into the diffusion model, and (*ii*) dynamic watermarks into images generated by the model. The fixed QR-code watermark uniquely encodes model-specific metadata, including the IP address of the training machine and timestamp information.

Our dynamic watermarking process enhances robustness by dynamically adjusting watermark transformations based on generated content within both feature and pixel spaces. In the feature space, high-level representations are extracted from original and watermarked images using a pre-trained feature extractor. We then calculate cosine similarity [28] between these features to quantify their semantic consistency. In the pixel space, image quality is evaluated using the Structural Similarity Index (SSIM) [1, 7, 8, 10, 13]. We propose a validation method that assesses the impact of blind watermarking on image quality through an analysis of 11 distinct image statistics.

2. Related Work

Fundamentally, diffusion models are generative frameworks inspired by diffusion processes in non-equilibrium thermodynamics. Typically, these models implement forward and reverse diffusion processes via finite Markov chains [19]. Recently, diffusion models have been adapted for conditional image generation tasks, such as image inpainting, text-guided generation, and editing. Their iterative denoising steps enable zero-shot image editing by guiding the generative process [22, 31, 38].

In generative models, watermark embedding within training datasets has been explored to protect the model's data [5]. However, this method can be inefficient, as adding

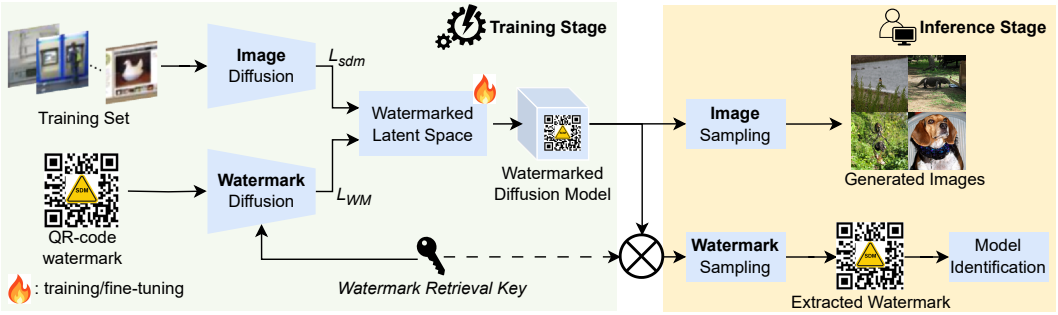


Figure 1. We embed a QR-code watermark into a target diffusion model, leveraging a watermark retrieval key to isolate watermark data and image data distributions. **(Left)** The training stage contains both watermark and image diffusion stages to construct a common, watermarked latent space. **(Right)** Sampling from the watermarked model will generate images as usual. When provided with the watermark retrieval key as an input in the sampling process, the watermark will be extracted, allowing for model identification

new watermark information typically requires costly retraining. Alternative approaches integrate watermark embedding directly into the generative process [20, 35], closely aligning with model watermarking techniques. Yet, such methods have two primary limitations: (*i*) they predominantly target GAN-based models, despite latent diffusion models (LDMs) increasingly replacing GANs in practical applications; and (*ii*) watermarks are embedded during initial model training [14, 26], which is resource-intensive and difficult to maintain. Recent studies indicate that efficient watermark embedding can be achieved through optimized fine-tuning of the generative model's latent decoder combined with an effective watermark extractor [15].

3. Proposed Method

We propose two watermark embedding branches: (*i*) a *model* watermarking branch that embeds a fixed QR-code watermark into the learned noise distribution of the diffusion model, ensuring robust traceability and ownership verification; and (*ii*) an *image* watermarking branch that embeds dynamic watermarks into generated images, balancing imperceptibility with resilience to attacks.

3.1. Model Watermarking

We employ the Stable Diffusion Model (SDM) [30] for our experiments. To enable watermark embedding, we modify the latent diffusion process by introducing a modified Gaussian kernel. The data distribution is defined as $q(z)$, where z represents data in the latent space. The watermark diffusion process serves as an extension of the traditional SDM diffusion process, introducing a watermark retrieval *key*, which we denote as 'κ' (see Fig. 1) to alter the diffusion pathway for the variable z_t such that:

$$\hat{z}_t = \gamma_\kappa z_t + (1 - \gamma_\kappa)\kappa, \tag{1}$$

where γ_κ is a blending factor that modulates the influence of the watermark on the generated output. Watermark embedding can be accomplished by fine-tuning the host model ϵ_θ^o. In each iteration, we sample a data instance z_0 from the training dataset D_{train} and a watermark example z_0^{w} from the watermark dataset D_{wm}. Noise samples ϵ and ϵ_{w} are then drawn from $N(0, I)$ separately for task and watermark data, along with a timestep t sampled from Uniform$(\{1, ..., T\})$.

For the watermark sample z_0^{w}, we first compute its latent representation z_t^{w} at timestep t within the latent diffusion process, and then construct the state $\tilde{z}_t^{\text{w}}$ in the Watermark Diffusion Process based on z_t^{w} as follows:

$$\tilde{z}_t^{\text{w}} = \gamma_\kappa \left(\sqrt{\bar{\alpha}_t} z_0^{\text{w}} + \sqrt{1 - \bar{\alpha}_t}\epsilon_{\text{w}}\right) + (1 - \gamma_\kappa)\kappa. \tag{2}$$

The joint learning optimization objective for the latent and watermark diffusion processes, which also serves as the loss function for watermark embedding:

$$L_{WDP} = E_{t\sim[1,T],z_0,z_{w0},\epsilon_t}\Big[\gamma_\epsilon\|\epsilon - \epsilon_\theta(z_t, t)\|^2 + \|\epsilon_w - \epsilon_\theta(\hat{z}_{wt}, t)\|^2\Big], \tag{3}$$

Here, γ_ϵ is a weighting factor that balances the standard diffusion process ϵ and the watermark diffusion process ϵ_w.

Watermark extraction can be achieved using the standard reverse diffusion process given the common latent space that was constructed through the combined diffusion processes discussed previously. The resulting output should be a reconstruction of the QR-code watermark that was used in the joint-training process.Given a model ϵ_θ, along with the known trigger κ and trigger factor γ_κ, we first sample z_T^{w} from $N(0, 1)$. Next, we calculate its corresponding state $\tilde{z}_t^{\text{w}}$ in the watermark diffusion process as follows:

$$\tilde{z}_t^{\text{w}} = \gamma_\kappa z_t^{\text{w}} + (1 - \gamma_\kappa)\kappa. \tag{4}$$

This state $\tilde{z}_t^{\text{w}}$ is then used as input to the model ϵ_θ to obtain the shared reverse noise to compute z_{t-1}^{w}. The extraction process follows the formula:

$$z_{t-1}^{\text{w}} = \frac{1}{\sqrt{\alpha_t}}\left[z_t^{\text{w}} - \frac{(1 - \alpha_t)}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_\theta\Big(\gamma_\kappa \hat{z}_t^{\text{w}} + (1 - \gamma_\kappa)\kappa, t\Big)\right] + \sigma_t z, \tag{5}$$

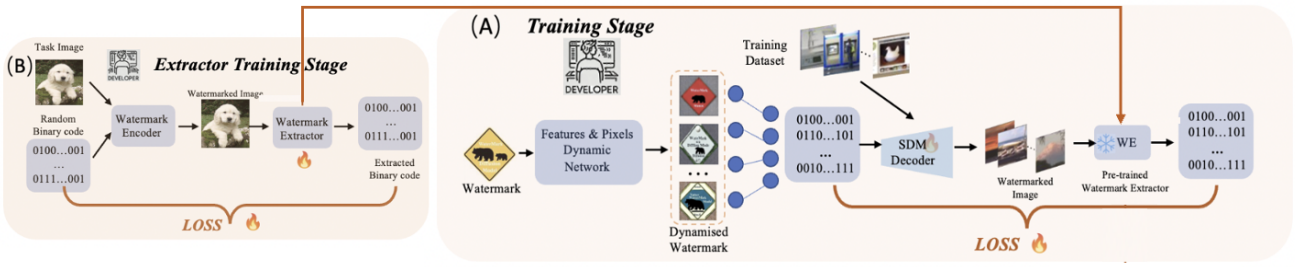


Figure 2. Overview of the dynamic watermarking framework, structured in three stages: (A) Training Stage: Fine-tunes the SDM decoder with a pre-trained watermark extractor to embed dynamic watermarks. (B) Extractor Training Stage: Trains the watermark extractor to encode and retrieve binary watermark information, maintaining reliability under transformations

where σ_t is the noise level at timestep t and z is a random sample from $N(0, I)$. The selection of the trigger key and the factor γ_κ should ensure sufficient divergence between the state distribution in the watermark diffusion process and that in the standard diffusion process.

3.2. Image Watermarking

Watermarks vary in shape and color while preserving key features of the reference watermark. In the feature space, we minimize the cosine similarity between original and transformed watermark features, enhancing orthogonality and robustness against extraction or tampering. Simultaneously, in the pixel space, we maximize the Structural Similarity Index Measure (SSIM) between original and transformed watermark images, ensuring high visual fidelity. These strategies enhance watermark robustness, imperceptibility, and classifiability without compromising generative capabilities.

To jointly satisfy constraints in both feature and pixel spaces, we formulate an optimization problem with the following objective:

$$\min\left(\lambda_{cosine} \cdot \cos\theta - \lambda_{SSIM} \cdot \text{SSIM}(I_o, I_t)\right), \quad (6)$$

where λ_{cosine} and λ_{SSIM} are weighting parameters that control the relative importance of cosine similarity and SSIM in the objective function. Original and transformed watermark images denoted as I_o and I_t.

In Fig. 2 (A), the watermark image is first converted into a binary sequence. We employ Huffman coding-based character encoding to reduce the size of binary sequence [18]. Watermark embedding consists of two steps. First, we train a watermark extractor. Then, we fine-tune the SDM decoder to embed specific watermark information in all generated images.

As shown in Fig. 2(B), the watermark extractor W_E training builds on HiDDeN [23]. We embed binary watermark information by jointly optimizing the parameters of the watermark encoder and W_E. Only W_E is retained as the watermark extractor. The watermark encoder takes a training image x_o and a binary watermark message m as inputs, producing a residual image x_δ. The watermarked image is obtained by scaling x_δ with a factor α. The extractor W_E then recovers the binary watermark m' as:

$$m' = W_E(T(x_o + \alpha x_\delta)) \quad (7)$$

In Fig. 2 Part (A), the SDM uses the latent vector decoded by decoder to produce a generated image. To support multiple watermarks, the decoder D_m is extended to accept both the latent vector z and the condition vector e_i, which specifies which watermark to embed. A training image is combined with the embedding vector e_i (derived from the condition i) to control the watermark embedding as follows:

$$x'_w = D_m(E(x) \in R^{h\times w\times c}, e_i) \quad (8)$$

The pre-trained extractor network W_E recovers the watermark m'_i from the generated image x'_w. The binary watermark information loss ensures that the extracted watermark m'_i matches the target m_i specified by the condition i:

$$L_m = -\sum_{j=1}^{k}\Big[{m_i}^{(j)} \cdot \log\sigma\left({m'_i}^{(j)}\right) + \left(1 - {m_i}^{(j)}\right) \cdot \log\left(1 - \sigma\left({m'_i}^{(j)}\right)\right)\Big] \quad (9)$$

4. Watermark Extraction Results

In row 1 of Fig. 3, our method effectively preserves the watermark's visibility and structure after extraction. The central logo undergoes minor deformation due to the diffusion process's inherent noise and transformations, which does not impact the QR code's scalability or embedded information. In row 2, our custom-designed "Diffusion Model" and "Watermark" serve as the core information. Each extracted watermark retains this text, while dynamic transformations modify its appearance. These transformations enhance robustness and resilience, yet the classifier can still accurately verify the watermark's origin.

Table 1. Image statistics comparison results. "Difference" refers to the percentage difference in data.Each Image statistics references:[34],[11],[12],[32],[32],[33],[21],[16],[6],[17], [8]

	GLCM Contrast	GLCM Energy	Canny Edge	Variance Blur Measure	Mean spectrum	Edge Histogram	Entropy Strength	Sharpness Score	Saturation	Texture	Image Realism
Watermarked	354.80	0.19	46.41	1124.47	85.73	6.00	6.31	8836.29	63.04	5.92	1.55
Clean	371.23	0.19	48.34	1209.43	88.18	7.00	6.23	8903.23	67.00	5.65	1.42
Difference	4.63%	0.00%	4.16%	7.56%	2.86%	16.67%	1.27%	0.76%	6.28%	4.56%	8.39%

Table 2. Classification results of Watermarked Images under various attacks. The classification network distinguishes whether an image contains a watermark and then classifies the watermark.

	Attack Type						
	No Attack	Rotation	Blurring	Texture Reduction	Image Compression	Crop	Flip
Watermark Presence	100.00%	99.30%	100.00%	95.70%	98.50%	99.10%	100.00%
Watermark Classification	97.00%	95.98%	93.97%	93.94%	94.72%	94.05%	96.10%

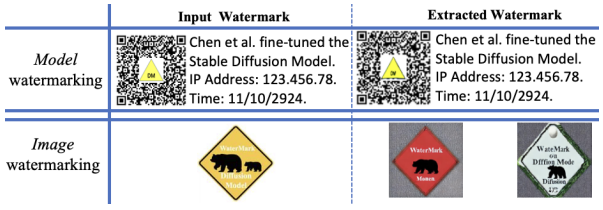


Figure 3. Raw 1 shows the extracted QR code watermark from the model watermark embedding branch. Raw 2 shows extracted watermarks from the image watermark embedding branch

We use image statistics based on human perception as evaluation metrics. Based on the sensitivity coefficient proposed by [8], we selected 11 image statistics measures with sensitivity values above 0.3, indicating a high sensitivity to changes in texture, edges, and frequency. Sensitivity ranges from 0 to 1, with higher values indicating greater sensitivity. As shown in Table 1, the average change between watermarked and clean images across these metrics is minimal, approximately 5%.

As shown in Table 2, texture reduction caused the largest drop (4.3%), likely due to its impact on watermark details, weakening detectability. Image compression, particularly lossy compression, also reduced accuracy to 98.50%. However, flip and crop attacks had minimal impact, maintaining accuracy at 100.00% and 99.10%, respectively. For the "Watermark Classification" task, attacks slightly reduced classification accuracy. Blurring and texture reduction had the greatest impact, lowering accuracy to 93.97% and 93.94%, respectively, suggesting these attacks degrade visual features, making classification more challenging.

To evaluate the effectiveness of our watermark embedding method in image generation, we compare our approach with three existing watermark embedding techniques using two widely used metrics: Inception Score (IS) [31] and Fréchet Inception Distance (FID) [22]. The IS measures the diversity and realism of the generated images, where higher scores indicate better quality. The FID assesses the distribution similarity between generated and real images, with lower scores indicating better quality. The comparison results are presented in Table 3. Our method outperforms the other approaches, achieving a higher IS and a lower FID, demonstrating its superior performance in maintaining image quality while embedding watermarks.

Table 3. Comparison of Image Generation Quality using IS and FID

Method	IS ↑	FID ↓
Tree-Ring [36]	3.80	3298.22
Chen et al. [9]	4.43	2789.98
Xin et al. [25]	4.14	3477.38
Clean images	5.01	2157.45
Ours	**4.61**	**2687.34**

5. Conclusion

In this paper, we propose a dual approach for embedding fixed QR-codes within the diffusion process and dynamic watermarks in generated images. This integration enhances intellectual property protection and traceability in generated content. The dynamic watermark undergoes feature and pixel space transformations, increasing resistance to attacks while preserving image quality. It remains intact even under rotation, blurring, and compression. Statistical validation shows dual watermarking method has minimal impact on image quality, with only a 5% variation in key metrics such as edge and texture attributes, confirming the method's invisibility and robustness. Our work addresses critical challenges in the ethical use of AI-generated content, providing a scalable and effective mechanism for ownership verification and misuse prevention. Future research could explore extending this framework to other generative models and applications, further advancing the field of digital security.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018. 1

[2] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 1

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 1

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. *OpenAI Technical Report*, 2024. 1

[5] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision*, pages 103–120. Springer, 2020. 1

[6] Kelvin CK Chan, Raymond H Chan, and Mila Nikolova. A convex model for edge-histogram specification with applications to edge-preserving smoothing. *axioms*, 7(3):53, 2018. 4

[7] Yunzhuo Chen, Naveed Akhtar, Nur Al Hasan Haldar, and Ajmal Mian. Deepfake detection with spatio-temporal consistency and attention. In *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2022. 1

[8] Yunzhuo Chen, Naveed Akhtar, Nur Al Hasan Haldar, Jordan Vice, and Ajmal Mian. A statistical image realism score for deepfake detection. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 1391–1396. IEEE, 2024. 1, 4

[9] Yunzhuo Chen, Jordan Vice, Naveed Akhtar, Nur Al Hasan Haldar, and Ajmal Mian. Image watermarking of generative diffusion models. *arXiv preprint arXiv:2502.10465*, 2025. 4

[10] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Proceedings of the 5th ACM workshop on information hiding and multimedia security*, pages 159–164, 2017. 1

[11] Kanjar De and V Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013. 4

[12] Huib de Ridder. Naturalness and image quality: saturation and lightness variation in color images of natural scenes. *Journal of imaging science and technology*, 40(6):487–493, 1996. 4

[13] Minh N Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. pages 2091–2106. IEEE, 2005. 1

[14] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. Supervised gan watermarking for intellectual property protection. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, Shanghai, China, 2022. 2

[15] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023. 2

[16] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020. 4

[17] Pedro Garcia Freitas, Luísa Peixoto Da Eira, Samuel Soares Santos, and Mylene Christine Queiroz de Farias. On the application lbp texture descriptors and its variants for no-reference image quality assessment. *Journal of Imaging*, 4 (10), 2018. 4

[18] Rishikesh R Gajjala, Shashwat Banchhor, Ahmed M Abdelmoniem, Aritra Dutta, Marco Canini, and Panos Kalnis. Huffman coding based encoding techniques for fast distributed deep learning. In *Proceedings of the 1st Workshop on Distributed Machine Learning*, pages 21–27, 2020. 3

[19] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992. 1

[20] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 2

[21] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. *Advances in neural information processing systems*, 18, 2005. 4

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 1, 4

[23] Chiou-Ting Hsu and Ja-Ling Wu. Hidden digital watermarks in images. *IEEE Transactions on image processing*, 8(1):58–68, 1999. 3

[24] Black Forest Labs. Flux.1. https://huggingface.co/black-forest-labs/FLUX.1-schnell, 2024. 1

[25] Xin Li, Xingjun Wang, Anqi Chen, and Linghao Xiao. A simplified and robust dct-based watermarking algorithm. In *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, pages 167–171. IEEE, 2017. 4

[26] Dongdong Lin, Benedetta Tondi, Bin Li, and Mauro Barni. Cycleganwm: A cyclegan watermarking method for ownership verification. *arXiv preprint arXiv:2211.13737*, 2022. 2

[27] Alistair Moffat. Huffman coding. *ACM Computing Surveys (CSUR)*, 52(4):1–35, 2019. 1

[28] Faisal Rahutomo, Teruaki Kitasuka, Masayoshi Aritsugi, et al. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, page 1. University of Seoul South Korea, 2012. 1

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 1, 4

[32] Bino Sebastian V, A Unnikrishnan, and Kannan Balakrishnan. Gray level co-occurrence matrices: generalisation and some new features. *arXiv preprint arXiv:1205.4831*, 2012. 4

[33] Taqi Tahmid and Eklas Hossain. Density based smart traffic control system using canny edge detection algorithm for congregating traffic information. In *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–5. IEEE, 2017. 4

[34] Du-Yih Tsai, Yongbum Lee, and Eri Matsuyama. Information entropy measure for evaluation of image quality. *Journal of digital imaging*, 21:338–347, 2008. 4

[35] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8695–8704, 2020. 2

[36] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023. 4

[37] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1

[38] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Durim Morina, and Michael S Bernstein. Hype: human-eye perceptual evaluation of generative models. 2019. 1

6. Supply Material

6.1. More Watermark Extraction Results



Figure 4. Extracted watermarks from the image watermark embedding branch. Each watermark has been dynamically transformed in shape and color.

The watermark's visibility and structure after extraction. In Fig. 3, our method effectively preserves the watermark's visibility and structure after extraction. The central logo undergoes minor deformation due to the diffusion process's inherent noise and transformations, which does not impact the QR code's scalability or embedded information. In Fig. 4,, our custom-designed "Diffusion Model" and "Watermark" serve as the core information. Each extracted watermark retains this text, while dynamic transformations modify its appearance. These transformations enhance robustness and resilience, yet the classifier can still accurately verify the watermark's origin.