VECTOR QUANTIZATION-BASED WATERMARKING FOR AUTOREGRESSIVE GENERATED IMAGES

Qiran Lai and Adrian G. Bors

Department of Computer Science, University of York, York YO10 5GH, UK

ABSTRACT

Watermarking is essential for identifying AI generated media, while ensuring their copyright protection and proper usage. Existing image generative AI watermarking methods are designed mainly forGenerative Adversarial Networks (GANs), or diffusion models, and not for AutoRegressive (AR) models. We propose the Vector Quantization based Watermarking for Autoregressive Generated Images (VQAGI-Mark), a self-attention based image watermarking approach, that selects visually similar token pairs from the codebook and embeds the watermark information through corresponding index replacements, while preserving image quality. Watermarking requires a challenging balancing among its requirements of invisibility, robustness, high bit embedding and security. The proposed watermarking approach embeds multiple bits while enhancing watermark' robustness. The multi-bit embedded watermark is resilient to a wide range of attacks. Experiments show that the proposed watermark achieves state-of-the-art autoregresive generative watermark fidelity and robustness for watermark identification while maintaining high visual quality. Code is available at `https://github.com/TuTAly/VQAGI-Mark`

***Index Terms*—** Generative AI watermarking, Autoregressive Image Generation Model.

1. INTRODUCTION

Autoregressive (AR) modelling emerged as a unifying paradigm across different modalities, achieving remarkable success in both natural language processing and visual generation tasks [1, 2, 3]. Based on the foundational principles of large language models (LLMs), recent AR-based image generation models such as LlamaGen [4] and the Visual AutoRegressive modeling (VAR) [5] have achieved good results. AR models employ a vector quantization (VQ) tokenizer [6] to discretize images into sequences of codebook indices, and the attention-based model for the "next-token prediction" task. This approach brings two notable advantages: higher inference efficiency compared to diffusion models, which are characterized by multiple denoising steps, and compatibility with the Large Language Models (LLM) frameworks, allowing them to benefit from shared architectures and pre-training paradigms [7, 8, 9].

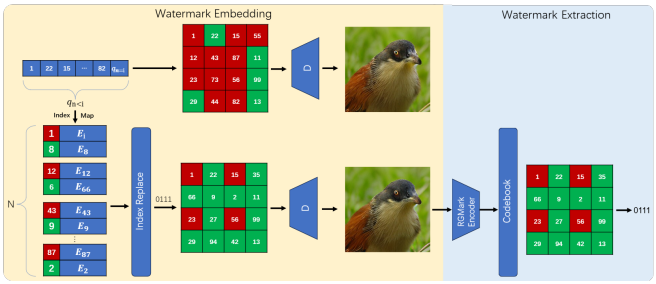


Fig. 1: The pipeline of watermark embedding and extraction of VQAGI-Mark, where the left part represents the watermark embedding process, while on the right is illustrated the watermark extraction process. Watermark embedding is performed via index replacement in paired codebooks. Red and green indices denote bits of 0 and 1.

AR image generation models have been shown to produce high-quality, high contrast, and photorealistic images which are increasingly adopted across various fields from creative industries to content platforms. However, this rapid advancement also raises serious ethical and legal concerns, including the fabrication of fake imagery, violation of the creators' content rights, and ambiguity in content ownership. The potential for misuse has triggered regulatory attention and societal scrutiny, with growing demands for developing traceability mechanisms that can identify the source of the generated images and enforce accountability while also preserving the copyright rights of their producers. Embedding invisible watermarks emerged as a promising solution for addressing such concerns. By embedding imperceptible signals directly into the generated image, watermarking allows model developers to trace image provenance, verify attribution, and support downstream auditing. Existing watermarking techniques can be broadly divided into two categories: post-processing methods [10], which add watermarks after image generation, and those embedded during the image generation, integrating watermarking with the image synthesis process. While post-processing watermarking is more straightforward to implement, if suffers from poor robustness while the watermarks can be easily removed. In contrast, embedding the watermarks through the generative pipeline is inherently more se-

cure and efficient.

Diffusion-based watermarking methods [11] are fundamentally different from AR models. The lack of parallelism, the token-level nature of AR decoding, and the reliance on discrete VQ codebooks introduce unique challenges for the watermark design in AR image generative models. Moreover, while existing works focus primarily on watermark verification (*i.e.* confirming the presence of a watermark), few methods address the more difficult problem of watermark identification—that is, attributing a generated image to a specific model or user via embedded signals, representing binary codes, robust to perturbations or attacks such as image cropping or compression.

In this paper we propose the Vector Quantization based Watermarking for Autoregressive Generated Images (VQAGI-Mark), a novel watermarking framework designed specifically for VQ-based image autoregressive generation models. The VQVAE model encodes an input image into a latent space that is discretized using a learned codebook of embeddings, then reconstructs the input by mapping discrete codes back through the codebook and decoder. VQAGI-Mark is based on the insight that VQ codebooks in AR models exhibit redundancy: many tokens are visually similar despite having distinct indices. This property enables us to embed watermarks by replacing the statistical distribution of token usage without perceptually altering the image. Specifically, VQAGI-Mark introduces a List Index Identification (LII) mechanism: the Red-Green List of embeddings are precomputed once via embedding similarity and divided into two sets. During image generation, the tokens are replaced with the watermark sequence. This novel replacement strategy effectively ensures embedding simultaneously multiple watermarking bits.

To enhance robustness, VQAGI-Mark further partitions the latent index sequence into multiple square spatial blocks that resist common image-level distortions such as JPEG compression, color jitter, Gaussian blur and random erasing. For the watermark identification, we reconstruct the token sequence from generated images via VQVAE encoding, compute the original index sequence, and identify the watermark following the majority voting rule in multiple blocks.

In summary, this paper brings the following contributions:

- We propose for the first time an AutoRegressive image generation watermarking model, which embeds watermarks through statistically controlled token replacement, leveraging codebook redundancy to find similar tokens. Our approach provides a practical and scalable solution for secure image attribution in AR-based generative systems.
- We consider an image block-partitioned embedding scheme for indices of images through optimized VQ codebook, improving robustness to a wide range of perturbations or attacks.

2. METHOD

LlamaGen [4], is a method representative of a new family of image generation models that employs the "next-token prediction" paradigm specific of large language models to the visual generation domain. It uses an image tokenizer and a scalable autoregressive image generation model that have been shown to generate images of high contrast and quality [4]. In the image AR model, tokens are generated and their relationships are then used to generate images through the VQVAE module. The pipeline of watermark embedding and extraction is shown in Figure 1.

Our watermarking process takes place in the quantized latent space of the LlamaGen VQ-based model. Firstly, the codebook index is represented into a sequence of red and green pairs corresponding to bits of zero and one, as shown in Fig. 1. Red and green indices are paired, according to their similarity in the codebook, to ensure low distortion in the generated watermarked image following index replacement. Then, given an input image, the encoder produces a sequence of quantization indices. The watermark bitstream is embedded by replacing selected indices according to the pairing rule: red indices represent bit 0, while green indices represent bit 1. Through this replacement, a subset of quantized indices is altered to embed the target watermark sequence. Finally, the replaced index map is then passed to the decoder to reconstruct the image, resulting in a visually indistinguishable output that implicitly contains the embedded watermark. The watermark embedding and watermark extraction method are shown in the left and right parts of Fig. 1.

2.1. Watermark Embedding

We embed watermarks into the tokens corresponding to the auto-regressive model. Each index corresponds to one of the two categories, shown as red or green in Fig. 1. We also use redundant square blocks to contain the same watermark information. Firstly, we divide all indices in the VQ codebook [6], $\mathcal{I} = \{1, 2, \ldots, N\}$, into two disjoint sets, referred to as Red-Green List. Specifically, let $\mathcal{I} = \{1, 2, \ldots, N\}$ denote the index set and corresponding partition of $\mathcal{I}$ into $N/2$ disjoint pairs $\{R_k, G_k\}$ and $k \in \{1, \ldots, N/2\}$.

To ensure that substitutions between paired Red-Green indices introduce minimal perceptual distortion, we construct the pairs by maximizing their embedding similarity. For any pairs, we calculate the cosine similarity between the indices in the codebook :

$$\mathrm{sim}(R_i, G_i) = \frac{\langle R_i, G_j \rangle}{\|R_i\| \, \|G_j\|}. \tag{1}$$

$$S_{\mathrm{sum}} = \max_{\mathcal{P}} \sum_{k=1}^{N/2} \mathrm{sim}(R_k, G_k). \tag{2}$$

and the final objective is to find an optimal assignment of all pairs that maximizes the total intra-pair similarity.

Computing pairwise similarities scales quadratically with N as $\mathcal{O}(N^2)$, which is computationally demanding. Therefore, to reduce the complexity of this procedure, we consider pairing as a maximum weight matching on a complete graph $Graph = (V, E)$, where each vertex corresponds to a codebook index and each edge weight is given by the cosine similarity between the connected embeddings. Since constructing and solving the matching on a dense complete graph is computationally prohibitive for a large N, we apply a Top-K pruning strategy: for each vertex, only the K edges with the highest similarity scores are retained, yielding a sparse graph that preserves the most promising candidate pairs. This pruning strategy reduces the overall computational complexity to $\mathcal{O}(NK)$. Noting that the optimization of $\mathcal{M}$ is a global one-time computational cost. In the experiments, we set Top-K pruning with $K = 10$. The maximum weight perfect matching on this pruned graph is then obtained using the Blossom algorithm [12], which efficiently handles general graphs by contracting odd cycles and guarantees an optimal assignment. This process results in $N/2$ disjoint index pairs with maximized intra-pair similarity :

$$\mathcal{M} = \{(R_1, G_1), \ldots, (R_{N/2}, G_{N/2})\}, \tag{3}$$

which maximizes the overall similarity. For each i, the paired index can be found through $\mathcal{M}(i)$, where i is the pair index. To ensure robust watermark identification, the image is firstly partitioned into a set of resolution-dependent blocks, which means that additional information can be embedded when generating higher-resolution images. The number of blocks is $H \times W/L^2$, where H and W are the height and width in the index map, while L is the hyper-parameter used to control the square side length of blocks.

2.2. Watermark Extraction

While the standard training process of the VQAGI-Mark lies in the context of image autoregressive generative model, it is designed to ensure the generated image reconstruction while embedding the watermark and its extraction aims to recover the bit sequence of the watermark while also preserving the generated image characteristics. Direct encoding and decoding of an image yields reconstructed indices $\widehat{\text{Idx}}^{rc}$ that may deviate from the original bit sequence Idx, sometimes leading to unreliable watermark detection. To address this, we enforce through deep learning training that the VQAGI-Mark encoder achieves a high-precision watermark extraction.

As illustrated in Fig. 2, we freeze the original VQVAE encoder and decoder, involved in the quantization codebook, and replace it with the VQAGI-Mark encoder to accurately recover the watermark token sequence because the original VQVAE encoder is only for image reconstruction while we also want to recover the embedded watermark tokens. Given an input image $\mathbf{I}^{real}$, the model encodes it into latent indices

Model	Method	Image quality				Accuracy ↑						
		PSNR ↑	SSIM ↑	CLIP ↑	FID ↓	Clean	Blur	Noise	JPEG	Bright	Erase	Avg
MSCOCO Dataset												
Post processing (256×256)	DwtDct	37.71	0.970	0.325	25.85	0.603	0.501	0.607	0.500	0.571	0.567	0.558
	DwtDctSvd	37.57	0.979	0.325	27.60	0.996	0.982	0.994	0.963	0.556	0.994	0.914
	RivaGAN	40.44	0.980	0.324	25.78	0.930	0.919	0.929	0.727	0.862	0.847	0.869
	W/o watermark	∞	1.000	0.328	26.55	–	–	–	–	–	–	–
AR Watermark (256 × 256)	IndexMark	23.54	0.838	0.326	24.73	1.000	0.991	0.995	0.978	0.988	0.997	0.992
	VQAGI-Mark	22.30	0.692	0.326	23.12	0.995	0.976	0.990	0.972	0.958	0.996	0.981
Post processing (512 × 512)	DwtDct	37.61	0.963	0.279	54.30	0.741	0.512	0.739	0.500	0.680	0.734	0.651
	DwtDctSvd	37.38	0.972	0.282	55.60	0.999	0.990	0.988	0.988	0.673	0.998	0.939
	RivaGAN	40.41	0.978	0.279	56.49	0.973	0.967	0.970	0.900	0.930	0.945	0.948
	W/o watermark	∞	1.000	0.403	25.53	–	–	–	–	–	–	–
Diffusion Watermark (512 × 512)	Tree-Ring	15.37	0.568	0.364	25.93	1.000	1.000	0.994	0.999	1.000	1.000	0.999
	ROBIN	24.03	0.768	0.396	26.86	1.000	1.000	0.998	0.971	1.000	1.000	0.995
AR Watermark (512 × 512)	IndexMark	24.15	0.838	0.281	54.35	1.000	0.988	0.994	0.984	0.989	0.992	0.991
	VQAGI-Mark	21.12	0.628	0.294	44.36	0.999	0.994	0.978	0.949	0.970	0.999	0.982
ImageNet Dataset												
Post processing (256 × 256)	DwtDct	38.73	0.974	0.288	15.17	0.583	0.501	0.588	0.500	0.584	0.568	0.554
	DwtDctSvd	38.44	0.979	0.288	15.32	0.994	0.991	0.989	0.960	0.552	0.994	0.913
	RivaGAN	40.44	0.980	0.288	15.29	0.951	0.930	0.950	0.746	0.919	0.914	0.902
	W/o watermark	∞	1.000	0.289	15.08	–	–	–	–	–	–	–
Diffusion Watermark (256×256)	Tree-Ring	15.68	0.663	0.267	17.68	1.000	0.994	0.999	0.998	0.798	0.995	0.964
	ROBIN	24.98	0.875	0.275	18.26	1.000	0.999	0.999	0.999	0.928	0.999	0.987
AR Watermark (256×256)	IndexMark	23.86	0.738	0.288	13.89	1.000	1.000	1.000	1.000	0.998	1.000	1.000
	VQAGI-Mark	22.54	0.633	0.326	18.73	0.999	0.971	0.962	0.952	0.949	0.952	0.964
Post processing (384×384)	DwtDct	39.36	0.972	0.286	12.50	0.720	0.521	0.725	0.500	0.780	0.696	0.703
	DwtDctSvd	39.08	0.979	0.286	12.56	0.990	0.992	0.990	0.966	0.664	0.999	0.863
	RivaGAN	40.45	0.977	0.286	12.79	0.966	0.947	0.964	0.846	0.949	0.939	0.935
	W/o watermark	∞	1.000	0.287	12.65	–	–	–	–	–	–	–
AR Watermark (384×384)	IndexMark	25.45	0.783	0.286	11.81	1.000	1.000	1.000	1.000	0.998	1.000	1.000
	VQAGI-Mark	23.54	0.638	0.260	11.73	0.997	0.991	0.995	0.978	0.988	0.997	0.991

Table 1: Comparison of different watermarking methods on MSCOCO and ImageNet datasets.

$\mathbf{z}$ and then reconstructs the image $\mathbf{I}$. Afterwards, the VQAGI-Mark Encoder recovers the indices $\widehat{\mathbf{z}}^{rc}$ and reconstructs the image $\widehat{\mathbf{I}}^{rc}$. The training objective minimizes two loss terms: (1) the mean squared error (MSE) between $\mathbf{z}$ and $\widehat{\mathbf{z}}^{rc}$, and (2) the MSE between $\mathbf{I}$ and $\widehat{\mathbf{I}}^{rc}$:

$$\mathcal{L}_{encoder} = \|\widehat{\mathbf{z}}^{rc} - \mathbf{z}\|^2 + \gamma\|\widehat{\mathbf{I}}^{rc} - \mathbf{I}\|^2, \tag{4}$$

where γ is a weighting hyperparameter. The first term ensures the recovery of the watermark, while the second ensures the generation of an image according to the prompt.

The sequence of binary values derived from all blocks forms the watermark. However, inconsistencies may occur due to possible distortions introduced during image processing. Therefore, we consider a redundancy-based voting scheme that improves the robustness against noise, image compression and other image distortions, thereby enhancing the watermark extraction reliability. The watermark extraction method is shown in the right part of Fig. 1.

3. EXPERIMENTS

We conduct watermarking experiments with the proposed Vector Quantization-based Watermarking for Autoregressive Generated Images (VQAGI-Mark) considering a high-quality autoregressive image generation model, called LlamaGen [4]. For the auto-regressive class-conditional image generation, we evaluate images synthesized at resolutions of 256×256 and 512×512. For text-conditioned generation tasks, we generate images at a resolution of 256×256 and 384×384. We consider embedding watermarks of capacity 16 bits so that all blocks contain 16 tokens, which means the block square side length is $L = 4$. All experiments are performed on standard benchmark datasets including COCO and ImageNet in order

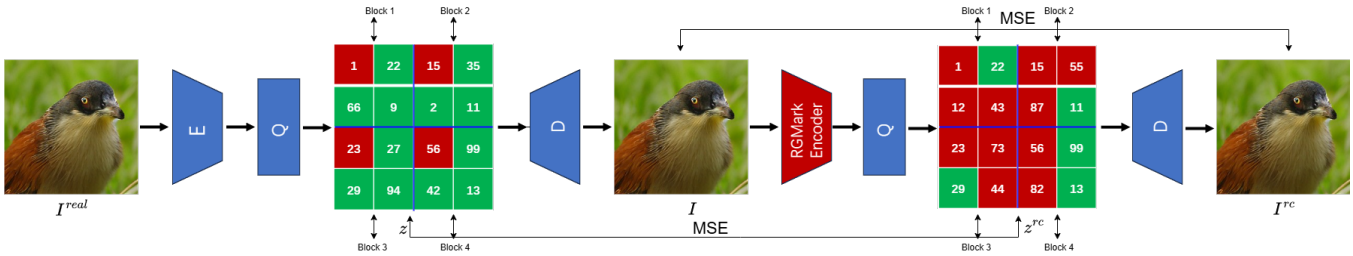


Fig. 2: VQAGI-Mark Encoder training process in watermark embedding.

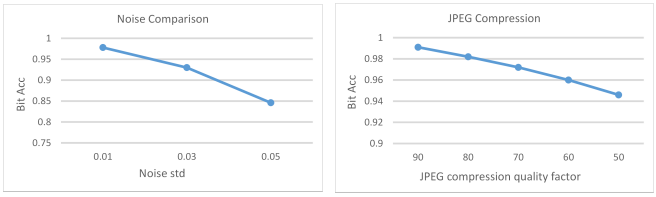


Fig. 3: Watermarked generated images with VQAGI-Mark considering labels from ImageNet as prompts in LlamaGen.

to compare with the results of other methods across evaluation metrics. To assess watermarking identification effectiveness, we use bit accuracy (ACC) as the primary measure. We also evaluate image fidelity using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [13]. Generated image distribution-level fidelity is evaluated using the Fréchet Inception Distance (FID) [14], and the semantic alignment between images and text prompts is measured with the CLIP score [15]. We compare our method against five baseline watermarking techniques: three post-processing approaches (DwtDct [16], DwtDctSvd [17], and RivaGAN [10]), two diffusion-based methods (Tree-Ring [18] and ROBIN [19]) and the Autoregressive watermark classification task [20].

For constructing index pairs, we apply Top-K pruning to control the range of optional index in the prediction by image AR model, which means only the Top-K tokens with the highest probabilities are retained and then renormalized before sampling. For class-conditioned generation, we use $K = 2000$, CFG (Classifier Free Guidance) scale of 4.0, and the downsampling size of 16, defaulting to the Red-Green watermark list. For text-conditioned generation, we use $K = 1000$, CFG scale of 7.5, and the same downsampling size of 16. The Index Encoder is trained with the Adam optimizer [21] using a learning rate of 1×10^{-5} and $\gamma = 0.5$. All experiments are conducted on an NVIDIA A40 GPU.

We show some a set of watermarked generted images in Fig. 3. Our experiments on COCO and ImageNet demonstrate the effectiveness of the proposed VQAGI-Mark compared with existing watermarking techniques. On both datasets, VQAGI-Mark achieves challenging watermark detection accuracy (average ≥ 0.9375, with only one missed bit) when considering a wide range of distortions, including image blurring, noise, JPEG compression, brightness adjustment. Although VQAGI-Mark exhibits lower image consistency compared with the generated image without watermark (*e.g.*, reduced PSNR/SSIM), its image quality is still good enough as shown in the FID and CLIP score, which means our watermarking method generates images similar to the original ones without watermark. Our method increases the watermark capacity compared with IndexMark [20] while still striking a good balance between robustness and imperceptibility. In Fig. 4-a we provide robustness test results when considering additive Gaussian noise, while in Fig. 4-b when increasing the JPEG compresion ratio (decreasing the quality factor).



(a) Additive Noise (b) JPEG compression

Fig. 4: Watermark bit accuracy when testing for attacks.

4. CONCLUSION

We propose a simple but effective strategy for watermark embedding and extraction in AI generated images by combining the quantization by VQVAE, binary mapping through image region classification, and a majority-voting correction scheme. This design enhances resilience against noise, image compression, blurring, and other distortions without relying on complex decoding architectures. Nevertheless, the approach has inherent limitations. The watermark capacity is rather low and it can only be increased with the generated image resolution. Also, by embedding watermarks into blocks of fixed size, we achieve a rather small robustness to geometric transformations such as cropping, rotation, or scaling. Future research will develop adaptive partitioning strategies and using error-correcting codes for further robustness improvement at various attacks.

5. REFERENCES

[1] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al., "PaLM 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al., "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al., "PaLM: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 1, pp. 11324 – 11436, 2023.

[4] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan, "Autoregressive model beats diffusion: Llama for scalable image generation," *arXiv preprint arXiv:2406.06525*, 2024.

[5] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang, "Visual autoregressive modeling: Scalable image generation via next-scale prediction," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, pp. 84839–84865, 2024.

[6] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural discrete representation learning," *Advances Neural Information Processing Systems (NeurIPS)*, pp. 6309 – 6318, 2017.

[7] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al., "Training compute-optimal large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, pp. 30016–30030.

[8] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al., "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[9] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

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

[11] Qiran Lai and Adrian G. Bors, "Localised frequency latent domain watermarking of ddim generated images," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[12] Jack Edmonds, "Paths, trees, and flowers," *Canadian Journal of Mathematics*, vol. 17, pp. 449–467, 1965.

[13] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[14] 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," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6629–6640.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*. PMLR 139, 2021, pp. 8748–8763.

[16] Ali Al-Haj, "Combined dwt-dct digital image watermarking," *Journal of Computer Science*, vol. 3(9), 09 2007.

[17] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker, *Digital watermarking and steganography*, Morgan kaufmann, 2007.

[18] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein, "Tree-rings watermarks: Invisible fingerprints for diffusion images," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024, pp. 58047–58063.

[19] Huayang Huang, Yu Wu, and Qian Wang, "ROBIN: Robust and invisible watermarks for diffusion models with adversarial optimization," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3937–3963.

[20] Yu Tong, Zihao Pan, Shuai Yang, and Kaiyang Zhou, "Training-free watermarking for autoregressive image generation," *arXiv preprint arXiv:2505.14673*, 2025.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.