ANCHMARK: Anchor-contrastive Watermarking against Generative Image Modifications

Minzhou Pan*†, Yi Zeng*[2], Xue Lin[1], Ning Yu[3], Cho-Jui Hsieh[4], and Ruoxi Jia[2]

[1]3Northeastern University, USA
[2]Virginia Tech, USA
[3]Salesforce Research, USA
[4]University of California, Los Angeles, USA

Abstract

This work explores the evolution of watermarking techniques designed to preserve the integrity of digital image content, especially against perturbations encountered during image transmission. An overlooked vulnerability is unveiled: existing watermarks' detectability significantly drops against even moderate generative model modifications, prompting a deeper investigation into the societal implications from a policy viewpoint. In response, we propose ANCHMARK, a robust watermarking paradigm, which remarkably achieves a detection AUC exceeding 0.93 against perturbations from unseen generative models, showcasing a promising advancement in reliable watermarking amidst evolving image modification techniques.

1 Introduction

Watermarks are crafted to uphold the integrity of digital image content origins and foster the fair use of images [1]. Over time, watermarks are refined to counter common perturbations like JPEG, rotation, or Gaussian noise often encountered during image transmission [2; 3; 4]. This resilience is expected to extend to emerging scenarios where image content is frequently modified by generative models, e.g., DALL·E 2 [5], Stable Diffusion [6], and Instruct-Pix2Pix [7]. However, our evaluation reveals an overlooked vulnerability of existing watermarks, as detectability may plummet to random guessing levels when faced with even moderate generative model modifications (Section 2).

To address the identified limitations and promote the fair use of image content, ANCHMARK is introduced: a robust watermarking paradigm inspired by contrastive learning. Engineered to anchor unperturbed samples, ANCHMARK learns a hidden space map of any watermarked counterparts post-perturbation close to the original watermarked content. This design necessitates only black-box access to potential perturbations, encompassing strong perturbations induced by computationally intensive generative models like the diffusion process, avoiding direct backpropagation of potential perturbations. Comprehensive empirical evaluations attest to ANCHMARK's stealthiness and robustness. Notably, against perturbations from unseen generative models, including DALL·E 2 image variation [5], ANCHMARK achieves a detection AUC exceeding 0.93. With 600 steps fine-tuning , ANCHMARK can achieve an AUC of 0.98 on DALL·E 2 image variation.

2 Background & Emerging Challenges

Diffusion Models. Recent advancements in diffusion models underscore their significance in enhancing image generation and editing quality [8]. By meticulously controlling noise addition and removal, these models transition from mere noise to refined images [9]. Notably, diffusion models offer the capability to condition the generative process, utilizing text or reference images to steer the output [10]. The advent of models like InstructPix2Pix [7] has broadened application horizons, facilitating text-guided image editing, varied object viewpoint production, and image inpainting

*Y. Zeng and M. Pan contributed equally.

Workshop on Regulatable Machine Learning at the 37th Conference on Neural Information Processing Systems (RegML @ NeurIPS 2023).

[11; 12; 13]. This capability presents an intriguing juncture with copyright law [1], propelling this study to ascertain reliable watermarking techniques supporting such regulatory objectives.

Revisiting Image Watermarking. Image watermarking embeds imperceptible but traceable information in images for copyright safeguarding. Traditional techniques employ hand-crafted watermarks in the frequency domain using transforms like DCT [2], DWT [3], and combined transforms (DctDwtSVD) [14]. Deep learning's advent ushered in refined watermarking methods, many adopting an encoder-decoder framework [4; 15], which enhances robustness through differentiable image distortion simulation. Recent innovations like Stable Signature (Stable Sig) [16] and Tree-Ring watermarks [17] directly manipulate the diffusion process, thus enabling watermarking generated contents from the modified diffusion models. We conduct a quantitive study of existing image watermarking techniques in the face of one of the most prevalent image variation tools, InstructPix2Pix [7]. Details of the experimental set-up and example of modification results are provided in Appendix D.

Table 1: Moderate-level InstructPix2Pix conditional modifications largely impact the detectability of existing watermarks. We set the hyperparameters of the InstructPix2Pix to a low level, with both text and image guidance scales set at 5. Humans can still easily pick up the connection between the modified version and the original image. We set the prompt used for modification in a randomized condition that includes "object change, style change, and background change," simulating the potential moderate level of potential image modifications. *Stable Sig and Tree-Ring cannot watermark real images or synthetic images that from models did not adopt their network modifications. We still included them here to showcase the prevalent vulnerability of existing watermarking techniques.

	DwtDctSVD [14]	**HiDDeN** [4]	**Stable Sig*** [16]	**Tree-Ring*** [17]
TPR @ 1% FPR	3.00%	4.20%	1.00%	13.30%
AUC	0.500	0.619	0.589	0.826

The results are collected in Table 1, where we evaluate the detectability in the format of the true positive rate at 1% of the false positive rate (TPR @ 1% FPR) and the area under the curve (AUC) for each watermark. From the results, we find even the most advanced watermarking techniques cannot maintain an acceptable detectability undergoing the perturbations introduced by the generative model, even though humans can easily connect the modified images and their origins.

Broader Impact. From a policy standpoint, the low detectability of watermarks carries implications extending beyond intellectual property infringement, as demonstrates by numerous real-life instances. Unsuccessful tracing of copyrighted content paves the way for potential financial and reputational damages, as creators find it challenging to assert rights [18]. This scenario further fuels unaccountable misinformation, with manipulated content masquerading as genuine, thereby amplifying fake news proliferation [19]. On an individual level, the ethos of consented image sharing is undermined as modifications and dissemination transpire without approval, leading to privacy violations [20]. Additionally, with the rising prevalence of synthetic data, compromised watermarks hinder the discernment of its origins, thereby contaminating the data ecosystem. Such contamination jeopardizes content authenticity and impairs future AI model performance via a phenomenon known as Model Autophagy Disorder, engendering a cascade of unreliable outputs [21].

3 Methodology

In this section, we present the methodology of our solution, termed ANCHMARK. Our approach embeds the watermark through the encoder network (E) and retrieves the watermark information via the decoder network (D). The watermark implanted in watermarked images is designed to withstand various image distortions, from traditional rule-based perturbations like JPEG to advanced ones like diffusion-based image modifications (P). The key difference between this work and the existing watermarking technique is that the robustness of watermarking is gradually learned through a contrastive-learning inspired process [22; 23], which does not require accurate backpropagation of the considered perturbations. The overall training process of ANCHMARK is illustrated in Figure 1 and comprises the following key components:

- **Watermark Encoder (E)**: Encoder embeds an invisible watermark into the original image, x, to produce its watermarked counterpart, x_w. The perceptual similarity between the watermarked image and the original image is quantified using the Visual Loss, denoted as $\mathcal{L}_v$ (Appendix A).
- **Perturbations (P)**: We randomly apply a list of perturbations (with random hyperparameters) to the original and watermarked images, i.e., x and x_w, accordingly producing the perturbed counterparts, x' and x'_w. We utilize a range of perturbations, including randomized generative model manipulations, the details of which are elaborated in Appendix B.

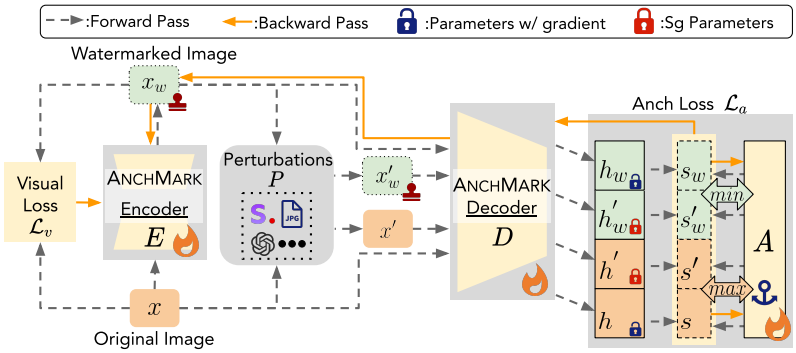

Figure 1: Training ANCHMARK. For stealthiness, the watermarked samples and the original images are used to compute a visual loss to ensure imperceptibility. For watermarking robustness, original images are watermarked, perturbed, and transformed into latent representations to calculate a combined Anch loss. Using the trained E and D, images are embedded with watermarks and assessed for watermark presence based on cosine similarity with the Anchor Vector A. *Sg means stop-gradient.

- **Watermark Decoder** (D): D maps images (x, x_w, x', and x'_w) into their latent representations. For instance, the original image x is transformed to $h = D(x)$. Similarly, the representations for x_w, x', and x'_w are given by h_w, h', and h'_w, respectively.
- **Anchor Vector** (A): This vector has the same dimension as the watermark decoder's output. The cosine similarity between A and each of the latent representations (h, h_w, h', h'_w) is computed, yielding similarity values s, s_w, s', and s'_w. These similarity measures are then input into the Anch loss, denoted as $\mathcal{L}_a$, to determine the final loss value.

As an example, the overall training process is as follows. We randomly sample an image, x, and forward pass through E to acquire the watermarked counterpart x_w. Then, we compare it with the original image x to calculate the visual loss, L_v. Meanwhile, the x and x_w are then passed through a set of random perturbations from P, including random image modifications introduced by generative models, producing x' and x'_w. This set of images, $\{x, x_w, x', x'_w\}$, are then decoded to their respective latent representations using the Decoder D. The cosine similarity between each latent representation and the Anchor Vector A is then computed. The final training loss is the summation of $\mathcal{L}_v$ and $\mathcal{L}_a$. The details of our loss function can be found in Appendix A. This combined loss is backpropagated to update the parameters of E, D, and A. It's important to emphasize that the parameters of A are updated solely using the non-perturbed views (h and h_w), since the strong image perturbations from P can introduce excessive randomness, making it challenging for A to converge [24]. Meanwhile, the design **does not require any backpropagation of the considered perturbations**, ensuring excellent generalizability to computational expensive perturbations (e.g., diffusion process) and even unseen perturbations that may present in the future.

To deploy ANCHMARK, we employ the trained E, D, and A. E is used to embed the watermark into the image. D then processes this image to yield its latent representation, subsequently employed to calculate the similarity with the Anchor Vector, A. The cosine similarity will be significantly higher for images containing the watermark (being perturbed or not), whereas non-watermarked images will exhibit a markedly lower value. To determine if an image contains the watermark or not, we evaluate the Bit Correct Ratio (BCR): $BCR = \frac{HD(\text{h,A})}{|\text{A}|}$, where HD is the Hamming distance [25]. A BCR of 1 indicates full accuracy, and 0 denotes a complete mismatch. Unlike other advanced watermarking techniques, i.e., Stable Sig [16] and Tree-Ring [17], **our method can be deployed to watermark any image**, including real images or synthetic data generated by black box generative models.

4 Evaluation

Set Up. We select representative watermarking techniques using different underlining mechanisms as our baselines. For traditional image watermarking, we chose DctDwtSVD [14]. For the encoder-decoder-based watermarking approach, Hidden [4] was selected. For diffusion model-specific watermarking, we choose two most recent works [16; 17] for comparison. We use the acquired BCR to compute AUC (↑) and TPR (↑) at 1% FPR as evaluation metrics. For the quantitative study, as detailed in Appendix D, we utilize 2,000 test images (as original images) in our evaluation. For watermark methods that can be deployed to any given image like ours, i.e., [14; 4], we embed the watermark into the 2000-size evaluation dataset and then introduce the perturbation based on the couped editing instructions using different generative models. For [16; 17], we generate the image using the image description provided with the 2000-size dataset and subsequently modify it based on the editing

instructions. As this work focuses on perturbations introduced by state-of-the-art generative models (which are largely centered on diffusion models), we omit the evaluation against traditional GANs. The considered image modification models include SDEdit [11], InstructPix2Pix [7], Zero 1-to-3 [12], InPaint [13], and the commercialized DALL·E 2 [5]. Note that [16; 17] cannot be deployed to watermark an existing image as the ones reside in the 2000-size dataset. We still include them as unfair baselines to emphasize the evaluation and comparison of the robustness of each watermark.

Table 2: Watermark detection performance against various image modification techniques. *Stable Sig and Tree-Ring's results are reflected from their own generated synthetic data.

	SDEdit[11]		InstructPix2Pix[7]		Zero 1-to-3[12]		InPaint[13]		DALL·E 2[5]	
	TPR (↑) @ 1%FPR	AUC (↑)	TPR (↑) @ 1%FPR	AUC (↑)	TPR (↑) @ 1%FPR	AUC (↑)	TPR (↑) @ 1%FPR	AUC (↑)	TPR (↑) @ 1%FPR	AUC (↑)
DwtDctSVD [14]	0.020	0.537	0.030	0.500	0.010	0.510	0.010	0.586	0.010	0.500
HiDDeN [4]	0.020	0.521	0.042	0.619	0.010	0.592	0.030	0.689	0.000	0.589
Stable Sig* [16]	0.010	0.510	0.010	0.589	NA		0.030	0.629	0.010	0.561
Tree-Ring* [17]	0.143	0.880	0.133	0.826	NA		0.218	0.895	0.231	0.856
ANCHMARK (Ours)	**1.000**	**1.000**	**1.000**	**1.000**	**0.910**	**0.990**	**0.854**	**0.966**	**0.713**	**0.938**

Experiments Result. From Table 2 we find that DwtDctSVD [14], HiDDeN [4], and Stable Signature [16] fail to detect the watermark after the evaluated image modifications, as the results are AUC that close to 0.5 – akin to random guessing. For DwtDctSVD [4], the non-detectability attributes to the most evaluated image modifications include the diffusion process, where high-frequency space is largely perturbed and reconstructed; thus, the watermark is distorted. For HiDDeN [4] and Stable Signature [16], their original design requires accurate computation of a linearized approximation of the considered perturbation; thus, they cannot be efficiently generalized to consider the perturbation introduced by nowadays generative models. Tree-Ring [17] embeds the watermark directly into the diffusion latent space and was reported by existing work indicating their robustness to imperceptible-level of discussion-based modifications [26]. In our evaluation, as we increased the modification scale, especially with designed prompt and guidance that simulates the reality level of moderate image modifications that users may introduce, we find that the TPR at 1% of FPR dramatically dropped to 23.1%. In contrast, ANCHMARK is able to maintain an AUC for all the evaluated perturbations. Note that we only used SDEdit [11] with randomly generated text (including gibberish prompts) instructions for the training process' considered perturbations (Appendix B). For unseen perturbation techniques introduced by other generative models and newly sampled instructions, ANCHMARK had no prior exposure. The outstanding robustness against unseen perturbations can be attributed to the inherent similarity of the evaluated generative models, which all deploy the diffusion process. Since ANCHMARK doesn't require model architecture details or gradient information, we can finetune trained E, D, and A directly on DALL E·2 to improve the performance of ANCHMARK customizely. In particular, we finetuned E, D, and A for 600 steps using around 11,100 new sampled DALL E·2 modified image pairs (approximated expense $180). After finetuning, ANCHMARK reach an AUC of 0.98 and a TPR of 0.824. This adaptability uniquely positions ANCHMARK to effectively handle perturbations introduced by unseen models and architectures. Further evaluations on visual comparison and traditional perturbations can be found in Appendix C.

5 Conclusion

This work identifies vulnerabilities in existing watermarking techniques against moderate image modifications using popular generative models. To address these weaknesses, we introduce ANCHMARK, a robust watermarking framework inspired by contrastive learning. ANCHMARK maximizes the cosine similarity between watermarked images and a trainable anchor vector in the decoder's hidden space, improving generalization against diffusion perturbations without direct backpropagation. Against unseen diffusion-model-based perturbations, ANCHMARK achieves AUC over 0.93, demonstrating resilience to generative models. By promoting data integrity amidst increasing synthetic content, ANCHMARK significantly advances reliable watermarking in our AI-driven landscape.

Acknowledgment

XL gratefully acknowledges the support of National Science Foundation Award No. CNS-1929300. RJ and the ReDS lab acknowledge support through grants from the Amazon-Virginia Tech Initiative for Efficient and Robust Machine Learning, the National Science Foundation under Grant No. IIS-2312794, NSF IIS-2313130, NSF OAC-2239622, and the Commonwealth Cyber Initiative. We want to thank anonymous reviewers for their valuable feedback and support.

References

[1] United States Congress. 17 u.s. code § 106 - exclusive rights in copyrighted works. U.S. Code, 2023. Available: https://www.law.cornell.edu/uscode/text/17/106.

[2] F.M. Boland, J.J.K. O'Ruanaidh, and C. Dautzenberg. Watermarking digital images for copyright protection. In *Fifth International Conference on Image Processing and its Applications, 1995.*, pages 326–330, 1995. doi: 10.1049/cp:19950674.

[3] Joseph JK O'Ruanaidh and Thierry Pun. Rotation, scale and translation invariant digital image watermarking. In *Proceedings of International Conference on Image Processing*, volume 1, pages 536–539. IEEE, 1997.

[4] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.

[5] 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.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[8] Tianyi Zhang, Zheng Wang, Jing Huang, Mohiuddin Muhammad Tasnim, and Wei Shi. A survey of diffusion based image generation models: Issues and their solutions. *arXiv preprint arXiv:2308.13142*, 2023.

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

[10] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

[11] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[12] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023.

[13] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[14] K. A. Navas, Mathews Cheriyan Ajay, M. Lekshmi, Tampy S. Archana, and M. Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 271–274, 2008. doi: 10.1109/COMSWA.2008.4554423.

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

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

[17] 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.

[18] Will J Francis. The two inescapable problems of ai art, Mar 2023. URL https://medium.com/@WillJFrancs/the-two-inescapable-problems-of-ai-art-6ef4d99ea97.

[19] Kayleen Devlin Cheetham and Joshua. Fake trump arrest photos: How to spot an ai-generated image, Mar 2023. URL https://www.bbc.com/news/world-us-canada-65069316.

[20] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023.

[21] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go mad, 2023.

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

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[24] Yalong Bai, Yifan Yang, Wei Zhang, and Tao Mei. Directional self-supervised learning for heavy image augmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16692–16701, 2022.

[25] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[26] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai, 2023.

[27] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *2014 22nd international conference on pattern recognition*, pages 34–39. IEEE, 2014.

[28] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[29] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

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

[31] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[32] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[34] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[35] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

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

A Loss function

As described in the previous section, our comprehensive loss function is defined as:

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_v \tag{1}$$

While $\mathcal{L}_a$ ensures that the decoder can effectively differentiate images with and without watermarks, $\mathcal{L}_v$ maintains the invisibility of the watermark.

Focusing on $\mathcal{L}_a$, the primary objective during training is to guide the decoder, D, such that it distances the latent representations of non-watermarked samples (h, h') from the anchor A, while drawing those of watermarked samples (h_w, h'_w) closer to A. Using cosine similarity as our distance metric, we aim for similarity measures s and s' to be close to -1, and s_w and s'_w to approach 1. We've adapted the binomial deviance loss [27] for our purposes. To simplify the expression, let's denote watermarked similarities s_w, s'_w as positive and represent them by s_i, while non-watermarked similarities s, s' are negative and represented by s_j:

$$\mathcal{L}_{\text{a}} = \sum_{i=1}^{m} \left\{ \underbrace{\tau \cdot \left(\log \left[1 + e^{(\lambda - s_i)/\tau} \right] + \log \left[1 + e^{(s_j - \lambda)/\tau} \right] \right)}_{\text{binomial deviance loss}} \cdot \underbrace{\text{sg} \left(e^{s_j - s_i/\tau} \right)}_{scaler} \right\} \tag{2}$$

The aforementioned loss is bifurcated into two components: the binomial deviance loss and a scaler. Delving into the binomial deviance loss, we introduce a temperature parameter τ, allowing the model to emphasize harder examples [28]. The margin λ is set to ensure a desirable separation between positive and negative samples. Transitioning to the scaler component, it gauges the disparity between positive and negative samples. When positive samples exceed negative ones in value, the scaler surpasses 1, thereby reducing the overall L_a and prompting the model to focus more on L_v. We also incorporate the temperature parameter τ to refine the scaling of the scaler. The term sg denotes the stop gradient, ensuring the scaler does not unnecessarily amplify computational overhead by extending the computation graph.

The second loss component, L_v, retains the image quality after its procession through the watermark encoder E. This loss function is articulated as:

$$\mathcal{L}_v = \alpha \mathcal{L}_{LPIPS} + \beta \mathcal{L}_{MSE} \tag{3}$$

In this context, we employ both LPIPS [29] loss and the MSE loss to capture visual deviations. The coefficients α and β serve as hyperparameters, adjusting the influence of each metric.

B Implementation details

Image Perturbations Our primary objective is to ensure robustness against multiple image Perturbations. To this end, we integrate a variety of perturbations during training. Taking advantage of the inherent properties of ANCHMARK—which eliminates the need for gradient propagation through these perturbations—we employ their original implementations directly. This approach guarantees both precision and operational efficiency. For each image, we randomly select a combination of one to three perturbations and combine them as P, which encompass JPEG, Gaussian blur, Gaussian noise, random rotations, brightness-contrast alterations, and the Diffusion-based image editing method, SDEDIT [11]. We initiate with mild Perturbations and gradually escalate their intensity, reinforcing our decoder's adaptability and promoting stable training.

Model Architecture. The adaptability of ANCHMARK enables compatibility with a diverse range of model structures. However, mindful of the practical requirements of real-world watermarking, we gravitate towards models with fewer parameters. This choice expedites both training and inference. Our encoder adopts a U-Net structure [30] with several SENet blocks [31], totaling a mere 0.22M parameters. The decoder harnesses the efficiency of MobileNet-V3 Large [32], streamlined to 2.5M parameters. We've transitioned its classification layer to a custom MLP to produce outputs congruent with the anchor vector A dimensions. Cumulatively, the entire encoder-decoder architecture is astoundingly lightweight at just 2.53M parameters, dwarfing even the basic model ResNet-18 [33] which houses 11.7M parameters.

Algorithm 1 PyTorch-style pseudocode of the ANCHMARK Training stage

```
# Initialize watermark Encoder and Decoder
E, D = encoder(), decoder(out_dim = anchor_len)

# Initialize Anchor Vector with ones
A = ones(1, anchor_len)

# Initialize optimizers for Encoder, Decoder, and Anchor Vector
opt_e, opt_d, opt_a = AdamW(E.params(), D.params(), A.params())

# Loop through the image and modify instruction using dataloader
for x, inst in dataloader:
    # Encode input data
    x_w = E(x)
    # Apply perturbation P with modify instruction to the data
    x_prime, x_w_prime = P((x, x_w), inst)
    # Decode the original and perturbed data
    h, h_w, h_prime, h_w_prime = D(x, x_w, x_prime, x_w_prime)
    # Compute cosine similarity with the anchor vector
    s, s_w, s_prime, s_w_prime = COS(A, (h, h_w, h_prime, h_w_prime))

    # Compute visual loss
    L_v = visual_loss(x, x_w)
    # Compute anchor loss
    L_a = anch_loss(s, s_w, s_prime, s_w_prime)
    # Compute total loss as sum of anchor loss and visual loss
    total_loss = L_a + l_v

    # Backpropagate the loss
    total_loss.backward()
    # Update the parameters using the optimizers
    opt_e.step(), opt_d.step(), opt_a.step()
```

Replace BN with GN. Image perturbations can drastically alter an image's statistics. For instance, changes in brightness directly modify pixel values, consequently altering the image's mean. Concurrently, the extensive size of the diffusion model restricts the training batch size. In our experiments, we were constrained to a relatively small batch size of 16. This combination of factors adversely impacts the performance of Batch Normalization (BN) [34]. So we replace the Batch Normalization (BN) with Group Normalization (GN) [35]. GN operates by normalizing groups of channels, eliminating the need for large batch sizes. This ensures stable and consistent training, even when image statistics vary widely.

C Additional Results

C.0.1 Watermark Visiual Comparision

	PSNR↑	SSIM↓	LPIPS↓
DwtDctSVD	**32.2197**	0.89598	0.10785
HiDDeN	30.8405	0.89548	0.12753
Ours	28.3564	**0.89521**	**0.04368**

Table 3: Comparison of visual similarity metrics for various watermarking methods. Higher PSNR values indicate less pixel-based difference between the original and watermarked images, whereas lower SSIM and LPIPS values suggest better preservation of structural and perceptual information, respectively.

The visual similarity between watermarked images and their original counterparts is essential to ensure that the watermark does not degrade image quality noticeably. For a clearer understanding, we

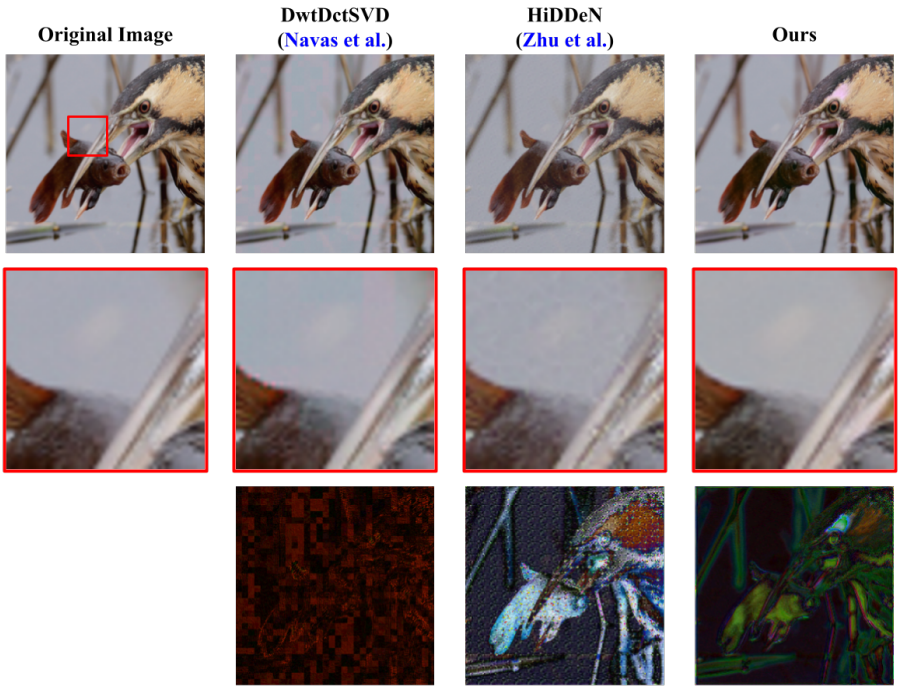

Figure 2: Visual comparison of different post-hoc watermarking techniques. Overall, local, and deficit ($\times 10$).

provide visual examples in Figure 2. Additionally, Table 3 quantitatively compares our method with other baseline techniques using three key metrics: PSNR, SSIM [36], and LPIPS [29].

From the results, DwtDctSVD achieves the highest PSNR value of 32.2197, indicating the least difference between the watermarked and original images in terms of pixel-based differences. However, it's worth noting that a higher PSNR doesn't always correlate to perceived visual similarity due to the non-linear nature of human visual perception.

Regarding the SSIM metric, which attempts to model the perceived change in the structural information of the image, all methods exhibit values close to each other. This suggests that the structural degradation caused by the watermarking process is relatively consistent across the techniques. Nonetheless, our method has the lowest SSIM of 0.89521, hinting at the least structural difference when compared to the other methods.

The most significant distinction arises in the LPIPS metric. Our method outperforms the other baselines with a score of 0.04368, nearly 50% lower than DwtDctSVD and almost 66% lower than HiDDeN. LPIPS is designed to be more aligned with human perceptual judgments, and a lower LPIPS value means that the difference between two images is less noticeable. Hence, the watermarks added using our method are less perceptually noticeable than those from the other two techniques, making our method superior in preserving the visual quality of the original image.

C.0.2 Robustness against Standard Image Perturbations

In our evaluation of the robustness of various watermarking methods against traditional image perturbations, as detailed in Table 4, we employ specific hyperparameters to simulate real-world scenarios. For JPEG compression, we used a quality factor (Q) of 80. When introducing Gaussian Noise, the noise standard deviation (σ) was set at 0.2. For Gaussian Blur, we utilized a kernel size (s) of 5 and a blur standard deviation (σ) of 1.5. To test the resilience against rotation, both random horizontal (H) and vertical (V) flips were applied. Finally, for alterations in contrast and brightness, pixels were manipulated by multiplying with a contrast factor, 'a', and subsequently adding 'b'. For our experiments, 'a' was set to 10, while 'b' remained fixed at 0.2.

With these parameters in mind, the analysis reveals that in the realm of JPEG Compression, most watermarking methods, especially AnchMark, display admirable resilience. Gaussian Noise, with

	JPEG		Gaussian Noise		Gaussian Blur		Rotation		Contrast & Brightness	
	TPR 1%FPR	AUC	TPR 1%FPR	AUC	TPR 1%FPR	AUC	TPR 1%FPR	AUC	TPR 1%FPR	AUC
DwtDctSVD [14]	0.962	0.989	0.280	0.841	0.806	0.963	0.542	0.913	0.243	0.684
HiDDeN [4]	0.972	0.997	0.482	0.903	0.774	0.958	0.937	0.998	0.802	0.964
Stable Sig [16]	0.770	0.955	0.330	0.858	0.740	0.949	0.883	0.984	0.582	0.911
Tree-Ring [17]	0.986	**0.999**	0.638	0.937	0.901	0.988	**0.988**	**0.999**	0.658	0.944
ANCHMARK (Ours)	**0.999**	**0.999**	**0.942**	**0.993**	**0.994**	**0.999**	0.984	**0.999**	**0.882**	**0.985**

Table 4: Comparison of watermark detection performance across various methods on traditional image perturbations. The table evaluates the True Positive Rate (TPR) at 1% False Positive Rate (FPR) and the Area Under the Curve (AUC) values for different image transformation models.

Category	Original Sentence	Edit Instruction
Object Change	The image features a close-up of a large crab...	Add another crab.
	The image features a close-up of a large, green lizard...	Remove the lizard.
	The image features a woman sitting on the grass...	Replace the dog with a cat.
	The image shows a person holding a black bag...	Change the black bag to a red bag.
	The image features a dog standing on a wooden floor...	Make the dog run.
Background Change	The picture features a mailbox sitting in a field...	Make the sky start raining.
	The image is a nighttime scene featuring a fish...	Replace the ground with a table.
	The picture features a man holding a black and white accordion...	Change the background color to green.
Style Change	The image features a man wearing a hat...	Turn it into an oil painting style.
	The image features a man riding a motorcycle...	Change the helmet's material to metal.

Table 5: Various examples are given to ChatGPT to generate Edit Instructions.

its defined standard deviation, was more challenging for most, yet our method excelled with a TPR of 0.942. Gaussian Blur, even with its significant kernel size and standard deviation, still saw our proposed technique achieving top results. Rotation's impact was notable, with most methods demonstrating strong resistance, especially when considering the added complexity of random flips. For alterations involving contrast changes, guided by our parameters, our technique maintained its exceptional performance. Overall, while several watermarking techniques showcased robustness against the perturbations with our chosen hyperparameters, our proposed method stood out, underscoring its robustness and adaptability in confronting a range of image perturbations.

D Expanded Experimental Set-Up

In this section, we further detail the procedure of experiment set-up and how we adapt the ImageNet [37] dataset with corresponding edit instructions for our evaluation.

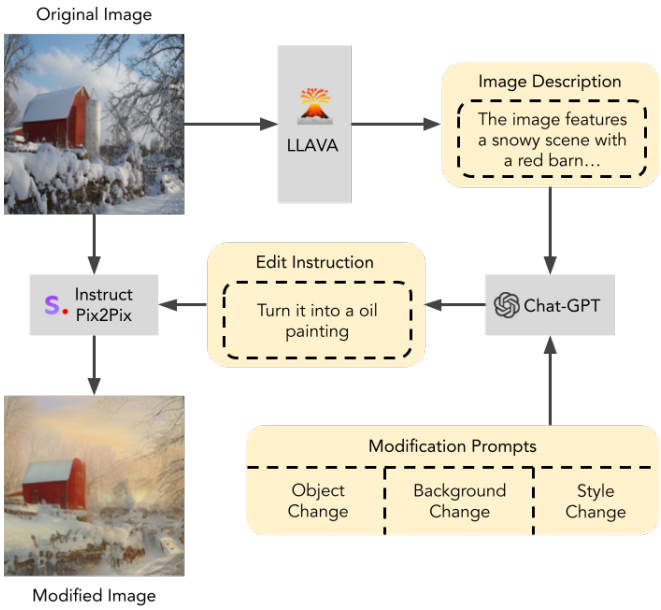

Figure 3: Workflow for the creation and implementation of the image edit instructions dataset.

In all evaluations (Section 2, 4), we keep the baseline method with their original implementation. For ANCHMARK, we set the latent space of Decoder D to 16-bits. For the evaluation dataset, a comprehensive visualization of this process is provided in Figure 3. The procedure begins with the original image, which is fed into LLAVA [38], a visual-language model. This model can respond to textual queries based on the provided image. By posing the question "What is the content of the image?" to LLAVA, it yields an approximate 60-word description of the image. This description, along with the modification prompts from Table 5, is then input into ChatGPT [39]. ChatGPT generates editing instructions based on this input. These instructions are subsequently fed into our diffusion-based image modification tool, Instruct Pix2Pix [7], to produce the final modified images.

Training Dataset for ANCHMARK. For the training dataset, we employ the previously mentioned method to generate editing instructions for the ImageNet [37] test dataset, which contains 100,000 images across 1,000 different classes. We shuffle the editing instructions for each image to simulate random perturbation, enhancing the model's robustness.

Evaluation Dataset. For the evaluation dataset, we also employ the previously mentioned method to generate editing instructions for the ImageNet [37] validation dataset, comprising 50,000 images from 1,000 different classes, selecting 1,000 images for our purposes. To enhance the diversity of the evaluation dataset, we incorporate an additional 1,000 images from the InstructPix2Pix dataset [7], maintaining each image with its original editing instruction.