WIBE: Watermarks for generated Images – Benchmarking & Evaluation

1st Aleksey Yakushev
ISP RAS
Moscow, Russia
yakushev@ispras.ru

2nd Aleksandr Akimenkov
ISP RAS
Moscow, Russia
alexandrakimenkov@gmail.com

3rd Khaled Abud
MSU AI Institute
Moscow, Russia
khaled.abud@graphics.cs.msu.ru

4th Dmitry Obydenkov
ISP RAS
Moscow, Russia
obydenkov@ispras.ru

5th Irina Serzhenko
MIPT
Moscow, Russia
i.f.serzhenko@gmail.com

6th Kirill Aistov
Huawei Research Center
Moscow, Russia
kirill.aistov@yandex.ru

7th Egor Kovalev
MSU
Moscow, Russia
egor.kovalev@graphics.cs.msu.ru

8th Stanislav Fomin
ISP RAS
Moscow, Russia
stanislav.fomin@gmail.com

9th Anastasia Antsiferova
ISP RAS Research Center, *MSU AI Institute*
Moscow, Russia
aantsiferova@graphics.cs.msu.ru

10th Kirill Lukianov
ISP RAS Research Center, *MIPT*
Moscow, Russia
lukianov@ispras.ru

11th Yury Markin
ISP RAS
Moscow, Russia
ustas@ispras.ru

***Abstract*—As invisible image watermarking gains importance for verifying AI-generated content, consistency and reproducibility remain major challenges due to the diverse methods, datasets, attacks, and metrics.**

We aim to provide a flexible, extensible, and user-friendly framework that enables systematic testing of watermarking methods under various conditions.

We developed WIBE, a framework with command-line interfaces and YAML configuration support, enabling users to evaluate a wide range of image watermarking algorithms on various datasets, apply configurable attack scenarios, and compute standard performance metrics. WIBE includes a library of pre-implemented methods and supports integration of new watermarking techniques, attacks, metrics, and datasets through a plugin-based architecture.

WIBE enables rapid prototyping, reproducible experiments, and insightful comparison of watermarking robustness. In our demo, we present its core features, plugin extensibility, and interactive infographics, making it a practical tool for researchers and practitioners working at the intersection of AI and media integrity.

Project on GitHub: https://github.com/ispras/wibe

YouTube video: https://youtu.be/lbWWB1crrwk

***Index Terms*—SE4AI, Invisible Watermarking, AI-generated Content Verification, Watermark extraction attacks, Trust AI**

I. Introduction

With the rapid rise of AI-generated imagery, ensuring the authenticity and integrity of visual content has become increasingly critical [1], [2]. Digital watermarking has emerged as a key technology for embedding imperceptible marks that verify the provenance of content and detect tampering [3], [4]. Yet, the evaluation of image watermarking methods remains a fragmented and challenging task [5]. Diverse embedding algorithms, heterogeneous attack models, varying datasets, and inconsistent metrics complicate reproducible benchmarking and fair comparison [4].

Existing frameworks and tools focus predominantly on watermark embedding or attack simulation, rarely supporting both domains comprehensively [6], [7]. Moreover, most solutions lack integrated visual analytics, which limits intuitive understanding of watermark robustness [7]. Additionally, many available repositories on platforms like GitHub provide collections of watermarking methods. Still, they mainly serve as static code dumps associated with publications, offering limited or no ongoing maintenance, and often lacking support for modern benchmarking practices [8], [9].

Critically, none of the existing tools simultaneously supports all three important features: interactive infographics for real-time analysis, scalability to run experiments on clusters, and integration of the latest SOTA watermarking methods [10], [11]. This gap hinders researchers and practitioners aiming to conduct systematic, reproducible, and insightful evaluations of watermarking approaches under realistic conditions.

To overcome these limitations, we developed WIBE: Watermarks for generated Images. Benchmarking & Evaluation, a flexible, extensible framework designed to unify the evaluation of image watermarking techniques across embedding and attack scenarios. WIBE empowers users to configure complex experiments through human-readable YAML files, execute evaluations on local machines or distributed clusters, and visualize results in real time to facilitate immediate insights

This work was supported by a grant provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4G0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated June 20, 2025, No. 139-15-2025-011. The research was carried out using the MSU-270 supercomputer of Lomonosov Moscow State University and ISP RAS computing cluster.

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00368

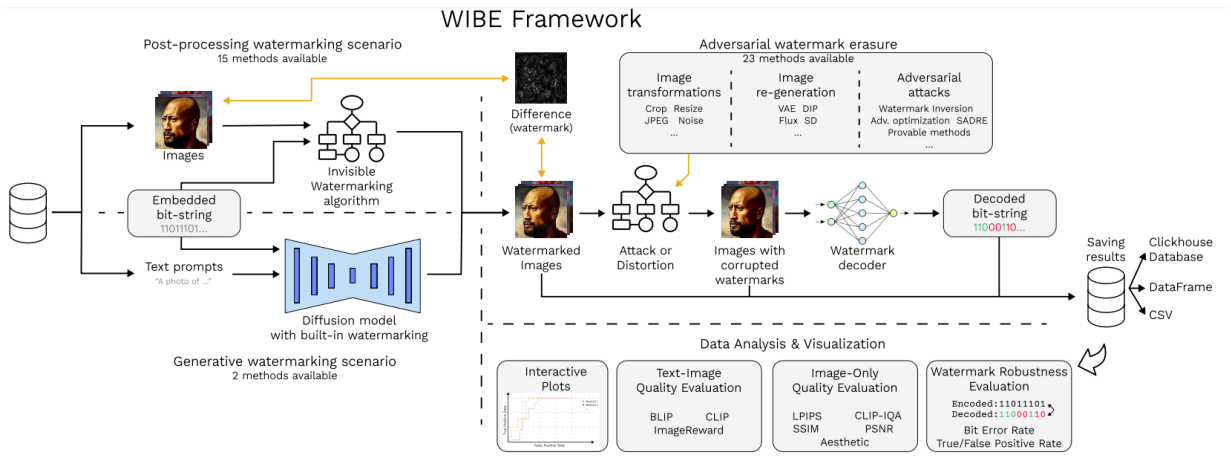

Fig. 1: Overview of the WIBE framework pipeline. The process begins with (1) Watermark Embedding and (2) Quality & Imperceptibility Evaluation on the left. The central module, (3) Attack Simulation, applies diverse perturbations and adversarial transformations. On the right, the pipeline continues with (4) Watermark Extraction, (5) Post-Attack Evaluation, and (6) Aggregation & Logging, while (7) Visualization & Reporting is performed as the final step (bottom-right).

(Table I).

Our framework is built around the following core principles:

1) Modularity and extensibility through a plugin-based architecture.
2) Reproducibility ensured by YAML-configured experiments.
3) Usability with a simple command-line interface.
4) Transparency via real-time visual feedback.
5) Scalability to run experiments on clusters.

The contributions of this work are as follows:

1) A unified framework supporting both watermark embedding and attacks.
2) Plugin architecture for easy extension with new methods, attacks, and metrics.
3) Real-time interactive infographics for intuitive performance visualization.
4) Scalability via cluster support for reproducible large-scale benchmarking.
5) Integration of state-of-the-art watermarking techniques alongside classic baselines.

II. System Overview

We present a modular and extensible framework for automated benchmarking of invisible image watermarking methods under various attack scenarios. The system is designed to support research and development of robust watermarking techniques by enabling systematic evaluation across a customizable pipeline. Its architecture comprises a sequence of processing stages (Figure 1), each of which can be configured via CLI, YAML configuration files, or Python-based backend scripts.

The pipeline consists of seven main stages:

1) **Watermark Embedding**: In the initial stage, an invisible watermark is embedded into an image using a selected method. The framework supports both post-generation embedding and embedding during the image generation process.
2) **Quality and Imperceptibility Evaluation**: The watermarked image is evaluated to ensure that the watermark remains invisible while preserving image quality. This stage supports multiple standard metrics (e.g., PSNR, SSIM) and allows integration of user-defined evaluation functions.
3) **Attack Simulation**: The system simulates a variety of attacks targeting the extraction and integrity of the watermark. These include adversarial techniques and standard perturbations, such as noise injection, compression, geometric transformations, and filtering. The attack stage is modular and extensible, enabling flexible configuration of both predefined and custom attack pipelines. In the context of watermarking, **robustness** refers to the ability of a watermark to remain reliably detectable after such perturbations, whether they occur intentionally (adversarial removal) or unintentionally (standard processing like compression or resizing).
4) **Watermark Extraction**: The attacked image is passed through the designated decoder to attempt extraction of the embedded watermark. Decoding performance under attack provides critical insight into the resilience watermark.
5) **Post-Attack Evaluation**: This stage assesses both the extracted watermark (e.g., accuracy, bit error rate) and the visual quality of the attacked image. The goal is to quantify success of the attack, with a focus on maximizing watermark disruption and minimizing image degradation. This step also supports custom evaluation metrics.
6) **Aggregation and Logging**: All intermediate and final results, including metric scores, images, and extracted

TABLE I: Comparison of frameworks for watermarking research. Symbols indicate feature support: ✓— full support; ⚠— limited support; ✗— no support. An asterisk (*) next to the number of watermarking methods denotes that the count is based on the methods reported in the original paper for the framework. In the public repository (main branch), we found mainly decoder implementations rather than full watermarking pipelines, although additional implementations may exist in other branches or versions.

Name	WA Methods	Attacks	Metrics	Infographics	Cluster/Parallel	SOTA Support
invisible-watermark [12]	3	0	1	✗	✗	✗
invisible-watermark(gpu) [13]	3	0	1	✗	⚠	✗
SSL watermarking [7]	2	0	4	✗	✗	✗
SSL watermarking attacks [14]	0	6	4	✗	✗	✗
WAVES [6]	5*	20	9	✓	⚠	✓
DL-Watermarking [15]	1	0	2	⚠	✗	✗
WIBE (ours)	**17**	**23**	**10**	✓	✓	✓

data, are aggregated into a structured output format. This enables reproducibility and facilitates comparison across experiments.

7) **Visualization and Reporting**: The final component automatically generates visual summaries and performance reports across a set of experiments.

Each stage of the pipeline is implemented as a modular component with standardized interfaces. Four stages (Watermark Embedding, Quality and Imperceptibility Evaluation, Attack Simulation, and Post-Attack Evaluation) are fully extensible, allowing researchers to integrate custom methods and criteria without modifying the core system. The remaining stages (Watermark Extraction, Aggregation and Logging, and Visualization and Reporting) follow fixed implementations to ensure consistency and comparability. This design enables rapid experimentation, fair benchmarking, reproducible results, and improved interpretability through detailed logging and visualization.

III. Use Cases

WIBE is a research-oriented framework designed for a systematic and reproducible evaluation of invisible image watermarking techniques. While not intended for production deployment, WIBE provides a flexible and extensible testbed for academic and applied research at the intersection of AI, media integrity, and security.

Core research use cases include:

- **Comparative evaluation:** Providing standardized pipelines and metrics to compare watermarking techniques across datasets, tasks, and threat models.
- **Method development:** Supporting rapid prototyping and debugging of novel watermarking algorithms and detection schemes via a modular plugin interface.
- **Attack simulation and robustness benchmarking:** Enabling controlled experimentation with custom or combined attack strategies to assess method resilience.

Beyond research, WIBE supports broader educational and engineering objectives:

- **Reproducibility and open evaluation:** WIBE encourages reproducible research by offering YAML-based experiment configuration and CLI automation, facilitating peer review and comparative studies.
- **Extensibility for benchmarking competitions:** Due to its plugin-based design, WIBE can serve as a standardized platform for shared benchmarks, leaderboards, and community-driven evaluation.
- **Education and outreach:** WIBE can be used in academic settings to teach students about digital watermarking, adversarial robustness, and multimedia forensics through hands-on experimentation.

To illustrate the capabilities of our framework for watermarking research, we conducted a small-scale experiment demonstrating how the system can be used for benchmarking and robustness evaluation across multiple watermarking methods and attack types. We selected 25,000 images from the DiffusionDB [16] dataset and embedded watermarks into each using four watermarking methods. After watermark insertion, we computed imperceptibility metrics to assess how visually undetectable the watermarks remained. Next, we subjected the watermarked images to different types of attacks. For each attacked image, we then measured the watermark detection success rate.

In this experiment, we selected four representative watermarking techniques: StegaStamp [17], SSL Watermarking [7], Trustmark [18], and Watermark Anything (WAM) [11]. Each method embeds an imperceptible watermark into image data. We evaluated the imperceptibility of these watermarks using several metrics across a subset of images, including PSNR, SSIM, LPIPS [19], and CLIP-IQA [20]. Table II summarizes the average perceptual distortion introduced by each method, indicating that all remain within acceptable visual thresholds.

TABLE II: Imperceptibility metrics after watermark embedding. ↑/↓ indicate desirable increase/decrease.

Method	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑
StegaStamp	29.133	0.920	0.052	0.813
SSL WM	42.140	0.973	0.043	0.815
TrustMark	44.178	0.994	0.001	0.853
WAM	41.370	0.988	0.017	0.821

To test robustness, we applied seven attack strategies to the watermarked images: JPEG compression with quality 50, 30 degree counterclockwise rotation, Gaussian blur with kernel size 5, Gaussian noise with standard deviation 0.05, center crop of half image area, BM3D [21], and DIP [22]. For each attack, we measured how well the watermark survived by computing the True Positive Rate at 0.1% False Positive Rate in the

TABLE III: TPR@0.1%FPR watermark detection performance after different attacks. A higher TPR value indicates that the watermarking method successfully resists the attack, while a lower value means that the attack has effectively disrupted or removed the watermark. Overall, there is no single watermarking method that remains robust against all types of attacks, and similarly, no attack can universally compromise all watermarking techniques.

Attacks	StegaStamp	SSL WM	TrustMark	WAM
JPEG	1.0	0.22	0.99	0.98
Rotate	0.0	0.91	0.0	0.0
GaussianBlur	1.0	0.98	1.0	1.0
Noise	1.0	0.08	1.0	1.0
CenterCrop	0.0	0.84	0.12	0.98
BM3D	1.0	0.0	1.0	1.0
DIP	1.0	0.43	0.99	0.17

watermark detection task (TPR@0.1%FPR). Table III presents the TPR@0.1%FPR results for all combinations of methods and attacks. As shown, no single method consistently resists all attacks, and conversely, no attack completely breaks every watermark. This highlights the importance of benchmarking watermarking approaches across various threat scenarios.

This experiment demonstrates how our framework supports reproducible and modular evaluation workflows for watermarking. Users can easily define evaluation pipelines, plug in new watermarking or attack modules, and collect standardized metrics.

IV. Conclusion and Future Work

WIBE addresses a critical need for systematic, reproducible, and extensible evaluation of invisible image watermarking methods in the face of rapidly evolving AI-generated content. Through its modular design, YAML-based configuration, and built-in library of methods, attacks, and metrics, WIBE simplifies comparative analysis, robustness benchmarking, and method prototyping. The demonstration experiment highlights that no single watermarking approach is universally robust, reinforcing the importance of comprehensive benchmarking under diverse threat scenarios.

By enabling seamless integration of new algorithms, attack strategies, and datasets, WIBE serves as both a practical research tool and a platform for community-driven benchmarking efforts. Its flexibility also makes it suitable for educational use, helping students and practitioners explore the complexities of watermarking and adversarial resilience. Moving forward, we envision WIBE fostering greater reproducibility, transparency, and collaboration in the study of digital media integrity.

References

[1] F. Ritchin, "When is a photo not a photo? the looming specter of artificially generated photographs," *Vanity Fair*, 2023.

[2] A. Sala, "Ai watermarking: A watershed for multimedia authenticity," *UN Agency for Digital Technologies*, 2024.

[3] F. Y. Shih, *Digital watermarking and steganography: fundamentals and techniques*. CRC press, 2017.

[4] L. Cao, "Watermarking for ai content detection: A review on text, visual, and audio modalities," *arXiv preprint arXiv:2504.03765*, 2025.

[5] K. M. Hosny, A. Magdi, O. ElKomy, and H. M. Hamza, "Digital image watermarking using deep learning: A survey," *Computer Science Review*, vol. 53, p. 100662, 2024.

[6] B. An, M. Ding, T. Rabbani, A. Agrawal, Y. Xu, C. Deng, S. Zhu, A. Mohamed, Y. Wen, T. Goldstein *et al.*, "Waves: Benchmarking the robustness of image watermarks," *arXiv preprint arXiv:2401.08573*, 2024.

[7] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou, and M. Douze, "Watermarking images in self-supervised latent spaces," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

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

[9] X. Xian, G. Wang, X. Bi, J. Srinivasa, A. Kundu, M. Hong, and J. Ding, "Raw: A robust and agile plug-and-play watermark framework for ai-generated images with provable guarantees," *Advances in Neural Information Processing Systems*, vol. 37, pp. 132 077–132 105, 2024.

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

[11] T. Sander, P. Fernandez, A. Durmus, T. Furon, and M. Douze, "Watermark anything with localized messages," *arXiv preprint arXiv:2411.07231*, 2024.

[12] ShieldMnt, "invisible-watermark," https://github.com/ShieldMnt/invisible-watermark, 2021, gitHub repository.

[13] S. AI, "invisible-watermark-gpu (v2)," https://github.com/Stability-AI/invisible-watermark-gpu, 2023, gitHub repository.

[14] V. Kinakh, B. Pulfer, Y. Belousov, P. Fernandez, T. Furon, and S. Voloshynovskiy, "Evaluation of security of ml-based watermarking: Copy and removal attacks," in *16th IEEE International Workshop on Information Forensics and Security (WIFS)*, Roma, Italy, December 2024.

[15] T. H. The, "Dl-watermarking," https://github.com/ThienHuynhThe/DL-Watermarking, 2022, gitHub repository.

[16] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, "DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models," *arXiv:2210.14896 [cs]*, 2022. [Online]. Available: https://arxiv.org/abs/2210.14896

[17] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2117–2126.

[18] T. Bui, S. Agarwal, and J. Collomosse, "Trustmark: Universal watermarking for arbitrary resolution images," *arXiv preprint arXiv:2311.18297*, 2023.

[19] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[20] J. Wang, K. C. Chan, and C. C. Loy, "Exploring clip for assessing the look and feel of images," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 2, 2023, pp. 2555–2563.

[21] Y. Mäkinen, L. Azzari, and A. Foi, "Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching," *IEEE Transactions on Image Processing*, vol. 29, pp. 8339–8354, 2020.

[22] H. Liang, T. Li, and J. Sun, "A baseline method for removing invisible image watermarks using deep image prior," *arXiv preprint arXiv:2502.13998*, 2025.