Pixel Seal: Adversarial-only training for invisible image and video watermarking

Tomáš Souček*, **Pierre Fernandez***, **Hady Elsahar**, **Sylvestre-Alvise Rebuffi**, **Valeriu Lacatusu**, **Tuan Tran**, **Tom Sander**, **Alexandre Mourachko**

Meta FAIR
*Equal contribution

Invisible watermarking is essential for tracing the provenance of digital content. However, training state-of-the-art models remains notoriously difficult, with current approaches often struggling to balance robustness against true imperceptibility. This work introduces Pixel Seal, which sets a new state-of-the-art for image and video watermarking. We first identify three fundamental issues of existing methods: (i) the reliance on proxy perceptual losses such as MSE and LPIPS that fail to mimic human perception and result in visible watermark artifacts; (ii) the optimization instability caused by conflicting objectives, which necessitates exhaustive hyperparameter tuning; and (iii) reduced robustness and imperceptibility of watermarks when scaling models to high-resolution images and videos. To overcome these issues, we first propose an adversarial-only training paradigm that eliminates unreliable pixel-wise imperceptibility losses. Second, we introduce a three-stage training schedule that stabilizes convergence by decoupling robustness and imperceptibility. Third, we address the resolution gap via high-resolution adaptation, employing JND-based attenuation and training-time inference simulation to eliminate upscaling artifacts. We thoroughly evaluate the robustness and imperceptibility of Pixel Seal on different image types and across a wide range of transformations, and show clear improvements over the state-of-the-art. We finally demonstrate that the model efficiently adapts to video via temporal watermark pooling, positioning Pixel Seal as a practical and scalable solution for reliable provenance in real-world image and video settings.

Correspondence: soucek@meta.com, pfz@meta.com
Website: https://facebookresearch.github.io/meta-seal
Code: https://github.com/facebookresearch/videoseal

1 Introduction

The rapid advancement of generative models such as DALL·E (Ramesh et al., 2022), Stable Diffusion (Rombach et al., 2022), Sora (Brooks et al., 2024), Veo (Google, 2025), and MovieGen (Polyak et al., 2024) has enabled the creation of high-fidelity synthetic content at scale. In response, invisible watermarking—the process of imperceptibly embedding a message into digital content—has emerged as a critical infrastructure for ensuring authenticity. This technique not only serves to distinguish synthetic media from real images, but also to establish broader provenance, such as verifying original uploaders and identifying source tools (Castro, 2025). With that, there is a pressing need for techniques that are both robust and imperceptible.

Multi-bit image watermarking, as established by the seminal work of Zhu et al. (2018), uses an embedder neural network to embed a binary message into an image as an imperceptible perturbation and an extractor neural network that retrieves the embedded binary message from the perturbed image. Typically, the embedder and extractor models are trained end-to-end by minimizing a compound loss function with two opposing objectives: message reconstruction loss to ensure the message can be recovered from a watermarked image, and perceptual losses such as MSE and LPIPS, or adversarial discriminator loss that ensures the embedded watermark remains imperceptible for humans. To achieve robustness of the embedded watermark against common user manipulations, such as application of Instagram filters, cropping, etc., during training, the watermarked image is augmented before the hidden message is retrieved using the extractor. By backpropagating through the augmentations, the model allows for the hidden message to be recovered even after image edits.

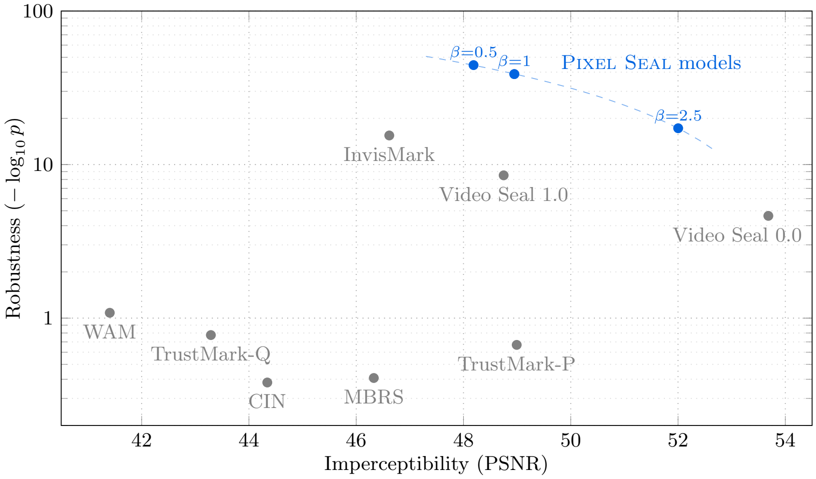

Figure 1 Imperceptibility and robustness of multi-bit image watermarking methods. The Pixel Seal family of models sets new state-of-the-art results for multi-bit image watermarking, both in watermark imperceptibility and robustness. The figure shows average values across 1000 test images generated by Meta AI. The robustness is measured for a combined attack of brightness change (0.5), crop (50% area), and JPEG compression (quality 40). Imperceptibility of Video Seal 0.0 is heavily skewed due to its small but very visible artifacts. Each Pixel Seal model is trained with a different watermark boosting factor β (see Section 4.1 for more details).

Despite the conceptual simplicity of this framework, training a model that is simultaneously robust, fast, and truly imperceptible remains notoriously difficult, as increasing the model's robustness can lead to more perceptible watermarks and increasing the model's imperceptibility often leads to less robust watermarks. We identify three fundamental bottlenecks in existing methods: First, standard training pipelines rely on a complex mix of perceptual losses to achieve watermark imperceptibility. These losses are often pixel-wise metrics, such as mean squared error (MSE), or deep perceptual metrics, such as LPIPS, but they remain imperfect proxies for human perception. For example, mean squared error loss does not discriminate between smooth and highly textured areas of an image, whereas humans are more likely to notice artifacts in the smooth regions of an image than in the highly textured ones. Second, jointly optimizing for robustness and imperceptibility creates a contradictory loss landscape—a global optimum for perceptual loss is a zero watermark (no message can be recovered), whereas a global optimum for robustness yields a highly visible watermark. Finding the optimal tradeoff between robustness and imperceptibility requires precise tuning of the learning dynamics. Without it, training frequently collapses—either the model fails to hide the information, or it hides it so well that the decoder cannot retrieve it. Third, applying existing models to high-resolution media at inference time requires watermark upscaling. This is due to the models being trained on low-resolution crops, and it results in distracting artifacts and inconsistencies that are easily visible to the human eye. In this work, we systematically address these challenges to advance the state of the art in image watermarking.

Our main contributions are as follows:

- First, we introduce an adversarial-only training paradigm that removes standard perceptual losses such as MSE or LPIPS. By relying solely on a discriminator, we avoid the failure modes and manual tuning associated with traditional loss functions.
- Second, we propose a three-stage training schedule that decouples the competing objectives of robustness and imperceptibility. By first achieving robust (but visible) watermarks and gradually enforcing invisibility, we ensure stable convergence across random initializations.
- Third, we propose to simulate the inference pipeline during training and apply Just-Noticeable Difference (JND) attenuation at the original input resolution. This eliminates the artifacts commonly found in upscaled watermarks while improving the robustness of the watermarks.
- Finally, we introduce Pixel Seal, an image watermarking model that achieves state-of-the-art robustness and imperceptibility (see Figure 1). We demonstrate that the model can be efficiently adapted to video via temporal watermark pooling with no performance drop.

2 Related work

Traditional methods. Early image watermarking methods operated either in the spatial domain (Van Schyndel et al., 1994; Nikolaidis et al., 1998) or in frequency domains like DFT (Urvoy et al., 2014), DCT (Bors et al., 1996; Piva et al., 1997), and DWT (Xia et al., 1998; Barni et al., 2001). In the video domain, watermarking approaches leveraged the specificities of video codecs to decrease the codec's impact on robustness. The resulting methods were typically designed for a specific video codec, such as MPEG-2 (Biswas et al., 2005; Noorkami et al., 2007) or H.264/AVC (Chen et al., 2006; Zhang et al., 2007; Mohaghegh et al., 2008).

Deep-learning based image methods have progressively replaced the traditional ones, as they make robustness easier by incorporating transformations directly into the training process. HiDDeN (Zhu et al., 2018), a seminal work in this direction, has inspired many extensions, through adversarial training (Luo et al., 2020), attention mechanisms (Zhang et al., 2020), and robust optimization (Wen et al., 2019). CIN (Ma et al., 2022) introduced invertible networks for the task of watermarking, while MBRS (Jia et al., 2021) focused on improving robustness to compression through the use of both real and an approximation of JPEG compression. More recent image watermarking works include resolution-independent embedding (Bui et al., 2023; Xu et al., 2025), robustness to diffusion purification (Pan et al., 2024; Lu et al., 2025), joint copyright protection and tamper localization (Zhang et al., 2024a), localized multi-message extraction (Sander et al., 2025), or very long hidden message length (Petrov et al., 2025). In the industry, Google's SynthID (Gowal et al., 2025) is deployed to watermark AI-generated content, whereas Meta (Castro, 2025) utilizes watermarking for a broader range of applications. Yet, details on these methods are scarce.

A parallel research direction focuses on watermarking AI-generated content during the generation process (Yu, 2020; Yu et al., 2021), with notable works including Stable Signature (Fernandez et al., 2023), Gaussian Shading (Yang et al., 2024), Tree-Ring (Wen et al., 2023), or their follow-ups (Kim et al., 2023; Hong et al., 2024; Ci et al., 2024) for diffusion models and WMAR (Jovanović et al., 2025) and BitMark (Kerner et al., 2025) for autoregressive models. In contrast to these works, we focus on watermarking in the *post-hoc* scenario, i.e., after the generation process.

Deep-learning based video methods are less numerous. Early works, such as VStegNet (Mishra et al., 2019) and RivaGan (Zhang et al., 2019), adapted HiDDeN to the video domain but faced efficiency challenges with 4D tensors. Recent works, like DVMark (Luo et al., 2023), VHNet (Shen et al., 2023), RC-VWN (Chen et al., 2024), or StegaVideo (Hu et al., 2024), have focused on improving robustness and imperceptibility through the use of video architectures, differentiable compression simulation, and embedding/extraction in the frequency domain (Zhang et al., 2024c; Chang et al., 2024), or tamper localization in videos (Zhang et al., 2024b). In contrast to these methods that rely on complex 3D architectures, ItoV (Ye et al., 2023) and Video Seal (Fernandez et al., 2024) adapt image-based architectures for video watermarking. Similarly, we also adapt Pixel Seal for video watermarking, but we propose inference time-only adaptation without the need for finetuning.

3 Preliminaries

In this section, we first describe a common post-hoc image watermarking framework, which will serve as a foundation for our Pixel Seal model described in the next section.

3.1 Neural-based image watermarking

A common approach for post-hoc multi-bit image watermarking is to train an embedder neural network that adds a binary message of length n_{bits} to an image and an extractor neural network that recovers it from the watermarked image. We detail their inference next, and describe the joint training process in Section 3.2.

Embedding and extraction of the watermark. The watermark embedder takes an image $x \in [0,1]^{H\times W\times 3}$ and a binary message $m \in \{0,1\}^{n_{\text{bits}}}$ as input. The output of the embedder is the watermarked image x_w or the watermark $w \in [-1,1]^{H\times W\times 3}$ which is then summed with the input image to produce x_w:

$$x_w = x + \alpha \cdot w, \quad w = \mathrm{Emb}(x, m), \tag{1}$$

where α is a scaling factor that controls the strength of the watermark. The watermark extractor takes a possibly altered version of the watermarked image x_w and outputs a "soft" message $\tilde{m} \in [0,1]^{n_{\text{bits}}}$ which is thresholded to recover the binary message $\hat{m} \in \{0,1\}^{n_{\text{bits}}}$.

Watermark detection with statistical guarantees. Multi-bit watermarking can be used for simple yes/no watermark detection in an image by consistently embedding the same binary message m and then verifying its presence in the image. A key advantage of this protocol is that it provides a statistical bound on false positives, which can be computed as:

$$\mathbb{P}_{m' \sim \mathcal{B}(0.5)^{n_{\text{bits}}}} \left[d_H(m, m') \leq d_H(m, \hat{m}) \right] = \sum_{k \leq d_H(m,\hat{m})}^{n_{\text{bits}}} \binom{n_{\text{bits}}}{k} 1/2^{n_{\text{bits}}} \tag{2}$$

where $\mathcal{B}(0.5)^{n_{\text{bits}}}$ denotes multivariate Bernoulli distribution and $d_H(\cdot,\cdot)$ is Hamming distance between binary messages. Equation (2) yields a theoretical error bound assuming, for any non-watermarked image, all binary messages are predicted with equal probability. This theoretical bound illustrates the benefit of multi-bit watermarking, which, unlike other approaches, does not require rigorous calibration on non-watermarked data to establish a reliable false positive rate.

3.2 Training image watermarking models

Commonly, during training, an image x is processed by the watermark embedder to produce the watermarked image x_w. This image is then augmented with various geometric, valuemetric, and compression transformations to simulate editing by users. Lastly, the augmented image $\tilde{x}_w$ is processed by the extractor to retrieve the original message hidden in the watermark. The models are trained end-to-end with a combination of various losses, as detailed next.

Losses. The post-hoc image and video watermarking training commonly use a combination of perceptual, adversarial, and message losses. The perceptual loss $\mathcal{L}_{\text{perc}}$ ensures the watermarked image x_w produced by the watermark embedder is close to the original input image x. The exact implementation of the loss varies significantly in the literature, ranging from simple mean squared error (MSE) to more complex ones, such as LPIPS (Zhang et al., 2018), focal frequency (Jiang et al., 2021), or Watson perceptual models (Czolbe et al., 2020). To improve the imperceptibility of the embedded watermark, many works rely on the adversarial loss $\mathcal{L}_{\text{adv}}$ often formulated as $\mathcal{L}_{\text{adv}} = -D(x_w)$. It maximizes the likelihood that a watermarked image x_w is classified as non-watermarked by the discriminator network D. The discriminator itself is jointly trained using a dual-hinge loss (Lim et al., 2017) to distinguish between original and watermarked images. Lastly, to hide a binary message m in the watermarked image, the binary cross-entropy loss $\mathcal{L}_{\text{msg}}$ is applied to the extractor outputs $\tilde{m}$:

$$\mathcal{L}_{\text{msg}}(m, \tilde{m}) = -\frac{1}{n_{\text{bits}}} \sum_{k=1}^{n_{\text{bits}}} m_k \log(\tilde{m}_k) + (1 - m_k) \log(1 - \tilde{m}_k). \tag{3}$$

The loss is backpropagated through both the extractor and the embedder to guide them to correctly embed and read the hidden message in the watermark. The final loss function used for the training is the weighted combination of the three losses:

$$\mathcal{L} = \lambda_{\text{perc}} \mathcal{L}_{\text{perc}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{\text{msg}} \mathcal{L}_{\text{msg}}. \tag{4}$$

Augmentations. To ensure the embedded message in the watermarked image can be read even if a user modifies the image, during training, the watermarked images are randomly transformed before being passed to the extractor. Common transformations include geometric changes (e.g., cropping, resizing, and rotation), valuemetric adjustments (e.g., brightness and contrast), and compression (e.g., JPEG). Some transformations, such as the JPEG compression, are not differentiable. In such a case, similarly to Zhang et al. (2021), this work uses a straight-through estimator that approximates the gradient with the identity function:

$$\tilde{x}_w = x_w + \text{nograd}(T(x_w) - x_w), \tag{5}$$

where T is the non-differentiable transformation. Other works approximate non-differentiable transformations with a trained neural network (Luo et al., 2023; Shen et al., 2023).

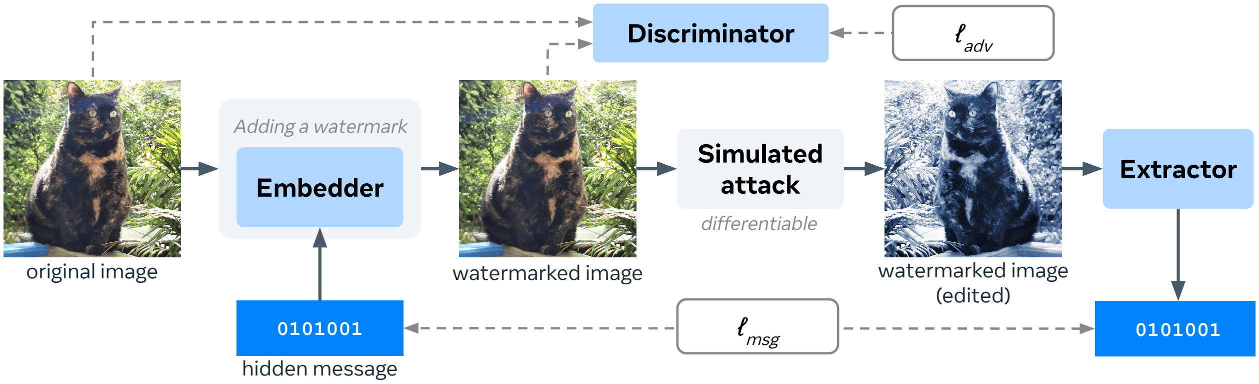

Figure 2 Our training setup. Pixel Seal is trained using only adversarial and message loss, which results in highly imperceptible watermarks. The message loss is backpropagated through the simulated attack (image augmentation) to ensure the embedded watermarks are robust to common user edits.

4 Invisible and robust image watermarking

Image watermarking presents several key challenges. First, the watermark must be invisible to humans; any visible artifacts, especially in flat regions such as the sky, are highly distracting to the viewer. Second, the watermark must be detectable even if the watermarked image is altered by various user edits, such as cropping or Instagram filters. Training such a robust model is highly sensitive to exact hyperparameter values, and multiple random initializations can yield significantly different outcomes. Lastly, watermarking must be fast and imperceptible even for very high-resolution images. A common inference-time interpolation technique for high-resolution watermarking introduces distracting artifacts that are easily detectable by the human eye.

To address these challenges, we introduce the following contributions. First, to produce invisible watermarks, we introduce adversarial-only training without any perceptual losses (Section 4.1). Second, to make the model robust and the training stable and repeatable, we introduce a three-stage training schedule in which we first achieve visible but very robust watermarks that are made invisible later during training (Section 4.2). Third, to allow for practical high-resolution watermarking, we propose training-time inference simulation to eliminate upscaling artifacts and increase model's robustness (Section 4.3). Lastly, we introduce temporal watermark pooling—a technique for efficiently adapting image watermarking models to video (Section 4.4).

4.1 Adversarial-only loss function

Ensuring that a watermark remains invisible in the content is the primary challenge in post-hoc watermarking research. For example, Bui et al. (2023) utilize four different losses specifically designed to make watermarks invisible. However, many such loss functions have modes in which the loss value is small, but the watermark is very visible to humans. Therefore, making the watermark invisible requires careful hyperparameter tuning without any guarantees of success. And even with carefully tuned hyperparameters, watermarks often remain visible in flat areas as waves (Bui et al., 2023) or blobs (Fernandez et al., 2024).

In the attempt to solve the imperceptibility issue, in contrast to previous work, as shown in Figure 2, we remove all perceptual losses and train the watermarking model using only the cross-entropy message loss $\mathcal{L}_{\mathrm{msg}}$ and an adversarial loss $\mathcal{L}_{\mathrm{adv}}(x, x_w)$:

$$\frac{1}{2}\nabla_{\theta_D}\Big[\mathrm{ReLU}\big(1-D(x)\big)+\mathrm{ReLU}\big(1+D(x_w)\big)\Big]-\lambda_{adv}\nabla_{\theta_{\mathrm{Emb}}}D(x_w) \qquad (6)$$

where x and x_w are the original and watermarked images, D is a patch-based discriminator network, θ_D its parameters, and θ_{Emb} are the parameters of the watermark embedder. The architecture of the discriminator affects how effectively it can remove visible artifacts. The more effective the discriminator is in removing visible artifacts, the less robust the final watermark is to various content modifications. In this work, we chose the patch-based discriminator of Rombach et al. (2022) as it provided a good balance.

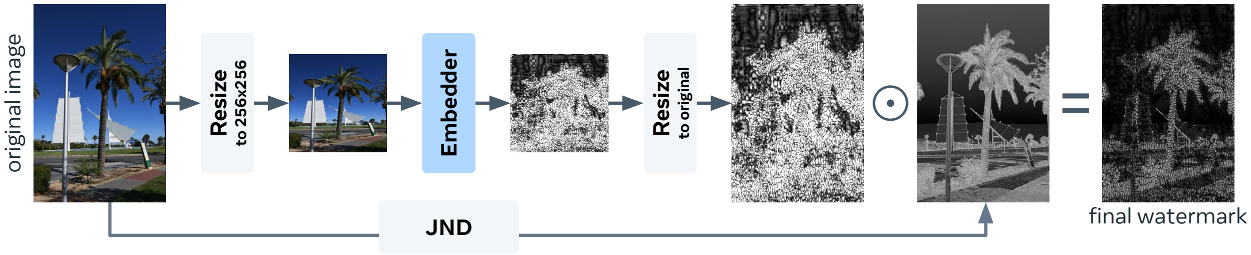

Figure 3 Training and inference of Pixel Seal on high-resolution images. The input image is first resized to the model resolution (256×256) and the raw watermark is computed using the Pixel Seal embedder. Then, this watermark is resized to the original input resolution and pixel-wise multiplied by the Just-Noticeable Difference (JND) map to obtain the final high-resolution watermark.

To control the imperceptibility of the watermark, we introduce the *watermark boosting* technique. The watermark boosting artificially amplifies the artifacts that are present in a watermarked image, making them easier for the discriminator to be detected and removed. In detail, the boosted watermarked image $\hat{x}_w$, computed as $\hat{x}_w = x + \beta(x_w - x)$, replaces the regular watermarked image x_w in Equation (6). Setting $\beta > 1$ ensures greater imperceptibility of the watermark, while $\beta < 1$ allows for very robust watermarks.

While the adversarial loss ensures the watermarked image does not contain any noticeable unnatural artifacts, the watermarked image can still significantly deviate from the original reference image. This is a lesser issue for watermarking of AI-generated content, where users are not provided with the original reference image, but it can be a major challenge for other provenance applications (Castro, 2025), such as professional photography devices, where the watermark must be minimal (ideally comparable to camera noise), so as not to alter the user's intent. To restrict the allowed deviation the watermark w can introduce to the original content, we reduce the watermark magnitude by a global scaling factor α and a local Just Noticeable Difference (Zhang et al., 2008) attenuation map as done by Sander et al. (2025). The attenuation map is computed using a non-learnable function $\text{JND} : [0,1]^{H\times W\times 3} \to [0,1]^{H\times W}$ which assigns large values for pixels where changes to pixel intensities will likely remain unnoticed by humans (e.g., edges) and small values otherwise. The watermarked image x_w is therefore computed as follows:

$$x_w = x + \alpha \cdot (w \odot \text{JND}(x)), \quad w = \text{Emb}(x, m). \tag{7}$$

4.2 Three-stage training schedule

Training the watermark embedder and extractor can be very brittle and sensitive to hyperparameters, as global optima of perceptual and adversarial losses correspond to an empty watermark $x_w = x$ ($w = \mathbf{0}$). Related works stabilize the training process by reducing the weight of perceptual losses λ_{perc} at the beginning of the training. Xu et al. (2025) even delay the application of augmentations τ to the later stages of training to improve the convergence of the model. Rather than carefully tuning hyperparameters, we observe a three-stage training can resolve the instabilities and allow the model to repeatably converge to a similar solution.

In detail, in the first stage, the model is trained only with the message reconstruction loss $\mathcal{L}_{\text{msg}}$ and the watermark scaling factor is set to a large value $\alpha = \alpha_0$, as done by Sander et al. (2025). At this stage, the predicted watermark is highly visible, yet it facilitates a stable learning process. We train the model until the bit accuracy is saturated, after which we move to the next stage. In the second stage, we add the adversarial loss $\mathcal{L}_{\text{adv}}$ and we gradually decrease the watermark scaling factor α from the initial value α_0 to the final value α_1 using a cosine schedule. We use the following schedule:

$$\alpha(t) = \alpha_1 + (\alpha_0 - \alpha_1) \cdot \cos\left(\frac{\pi}{2}\varphi\right), \quad \varphi = \text{clip}\left(\frac{t - N_{start}}{N_{end} - N_{start}}, 0, 1\right). \tag{8}$$

The factor φ controls the interpolation schedule based on the current epoch t, the second stage start epoch N_{start}, and the second stage end epoch N_{end}. The second-stage training ensures that the resulting watermark is imperceptible while still being robust against various attacks. In the final stage, we finetune the model with the final watermark scaling factor value $\alpha = \alpha_1$, potentially using different types of data, attacks, etc., to tailor the watermarking model for any specific application needs.

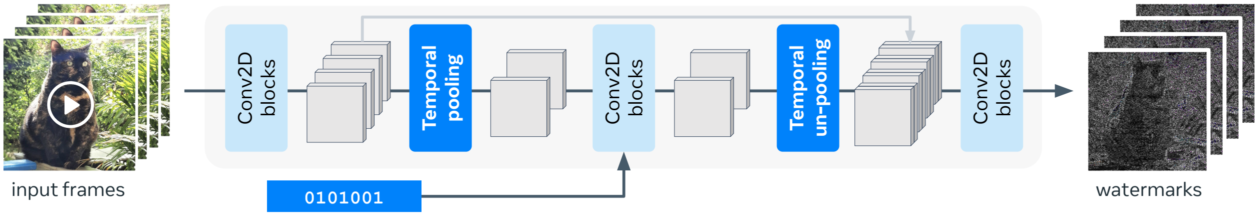

Figure 4 The embedder with temporal watermark pooling enabled. During inference, a temporal average pooling and un-pooling layer is inserted into the embedder. This modification results in a significant speedup for video watermarking, with no impact on imperceptibility or robustness.

4.3 High-resolution adaptation

Computing a watermark directly on a high-resolution image with millions of pixels is computationally prohibitively expensive. Therefore, a common approach is to train a watermark embedder and extractor using a fixed spatial resolution, e.g., 256×256, and use watermark interpolation technique (Bui et al., 2023; Sander et al., 2025) for inference. The technique works by downsampling the original image x to the fixed model resolution using bilinear interpolation and later upsampling the low-resolution watermark w back to the original resolution as shown in Equation (9).

$$x_w = x + \alpha \cdot \mathrm{resize}_{\uparrow}(w), \quad w = \mathrm{Emb}\left(\mathrm{resize}_{\downarrow}(x), m\right) \tag{9}$$

However, this approach has three major downsides. First, the watermark upsampling can often result in artifacts being spilled into regions where they are more visible (e.g., areas around edges). Second, some artifacts that are imperceptible in low resolution can be easily spotted in high resolution. And third, the watermark extractor is trained to extract hidden messages from images $\tilde{x}_w = \mathrm{resize}_{\downarrow}(x) + \alpha \cdot w$, where both $\tilde{x}_w$ and w are in the fixed model resolution, while during inference the extractor detects the hidden message in image $\tilde{x}'_w = \mathrm{resize}_{\downarrow}(x + \alpha \cdot \mathrm{resize}_{\uparrow}(w))$. Because in the general case $\tilde{x}_w$ and $\tilde{x}'_w$ are not equal, the watermark extractor is tasked with detecting watermarks in out-of-distribution data.

We address these challenges as follows: First, to address the spill of watermark artifacts into flat regions where they are more visible, we compute the JND attenuation map in the original image resolution. Although this increases the computational requirements during inference, the increase is substantially lower than the computation of the watermark in higher resolution. Second, we address the perceptibility of certain artifacts in higher resolution by computing the adversarial loss at the original image resolution. Although this increases training compute requirements, it has no effect on inference. Third, to eliminate the distribution shift of the watermark extractor input during inference, we simulate the same inference process during training. Again, this has no effect on inference. In detail, the training pipeline is outlined in Equation (10), with the key components illustrated in Figure 3.

$$\tilde{m} = \mathrm{Ext}(\tilde{x}_w), \quad \tilde{x}_w = \mathrm{resize}_{\downarrow}(\tau(x_w)), \quad x_w = x + \alpha \cdot \left(\mathrm{resize}_{\uparrow}(w) \odot \mathrm{JND}(x)\right), \quad w = \mathrm{Emb}\left(\mathrm{resize}_{\downarrow}(x), m\right) \tag{10}$$

The original image x and the watermarked image x_w are in the full resolution, while the watermark w and the input to the watermark extractor $\tilde{x}_w$, augmented by a random differentiable augmentation τ, are in the fixed model resolution.

4.4 Video watermarking through image model adaptation

Video watermarking presents two additional challenges compared to image watermarking. The first challenge is the complexity of video training itself—video decoding and pre-processing are resource-intensive, and the high correlation within adjacent video frames results in insufficient diversity within a batch. The second challenge is the computationally expensive process of watermarking each video frame during inference, which renders any application of a video watermarking model impractical, even with the resizing trick introduced in Section 4.3.

The simplest solution to the second challenge, i.e., increasing inference speed, is to watermark every k-th frame. However, this introduces distracting flickering if the watermark is not fully imperceptible, makes the watermark vulnerable to video compression, and requires exhaustive scanning during extraction since most frames are watermark-free. Therefore, Fernandez et al. (2024) propose to propagate the same watermark computed for every k-th frame to the remaining $k - 1$ frames. While fast, this approach introduces ghosting artifacts and requires video finetuning with the exact parameter k to make the method robust. Instead, we introduce *temporal watermark pooling*—an inference-time method applicable to our image watermarking model that alleviates the need to train on video while allowing for much faster inference without any loss in visual quality or robustness compared to the baseline of watermarking every frame.

The temporal watermark pooling, shown in Figure 4, is based on the observation that adjacent video frames are highly correlated. That is, the frames will likely have very similar high-level semantic features. Therefore, we propose replacing these per-frame high-level semantic features with their average across neighboring frames. In detail, during inference only, we insert a temporal average pooling layer with a kernel size and stride of k into our embedder after the d-th downsampling U-Net block. Similarly, we insert a temporal un-pooling (repeat) layer at the same position in the upsampling blocks of the U-Net. With d small, our temporal watermark pooling achieves a significant speedup while having no impact on imperceptibility or robustness due to the low-level per-frame changes computed using the last U-Net layers.

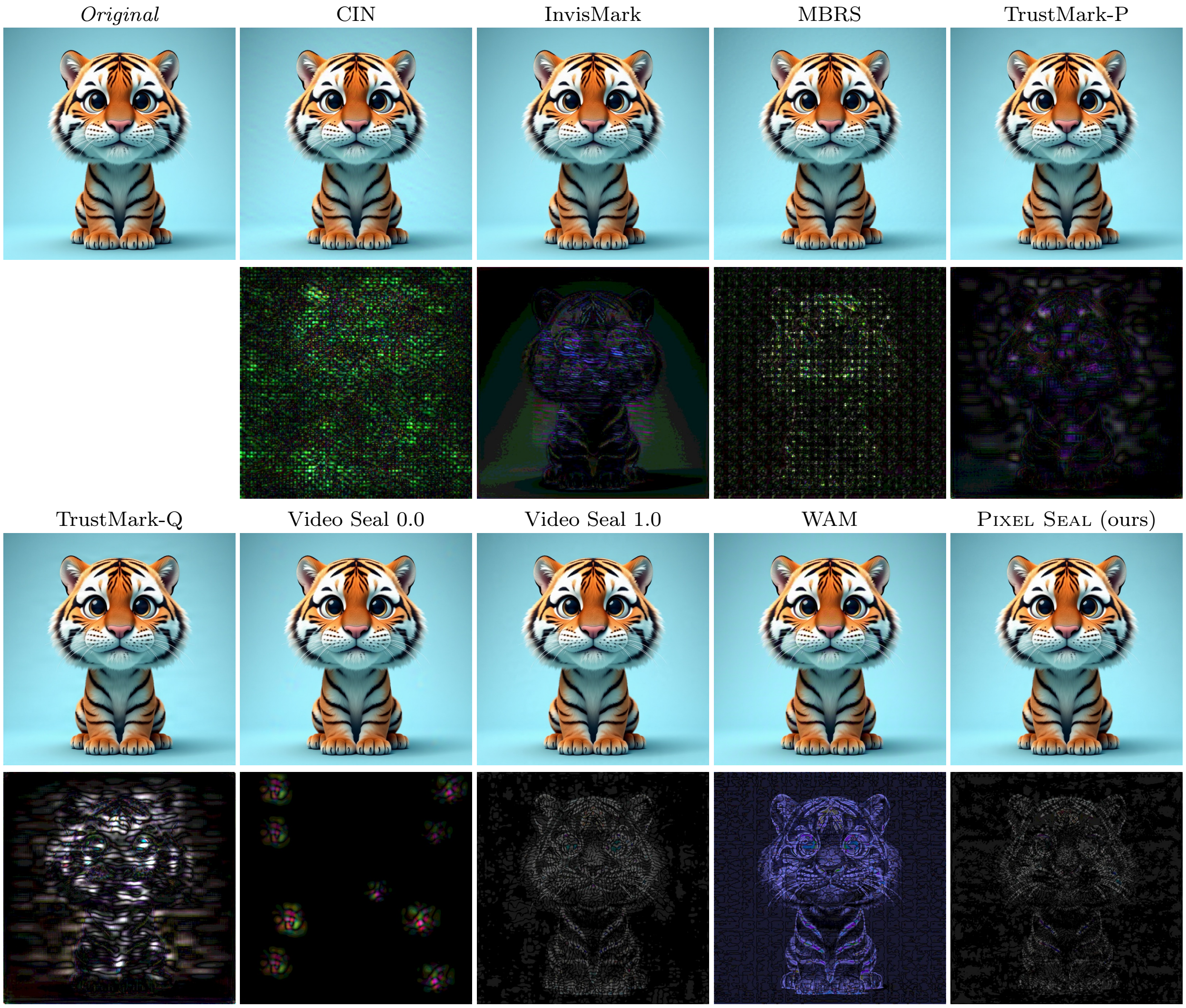

Figure 5 Comparison with related work on an AI-generated image. We show both the watermarked image (top) and the predicted watermark brightened for clarity (bottom). Many related methods leave visible artifacts in areas with a single color. In contrast, Pixel Seal does not leave visible artifacts in such areas while being more robust to various transformations. More examples are available in the appendix.

5 Experiments

In this section, we present the experimental setup and results of our image watermarking approach. We describe the implementation details and evaluate our method against state-of-the-art image watermarking models. Additionally, we conduct ablation studies to thoroughly verify our design choices.

5.1 Implementation and evaluation details

Implementation details. We train our model, called Pixel Seal, with a U-Net-based architecture (Saharia et al., 2022) with 43.8M parameters for watermark embedding and ConvNext-v2 Tiny (Woo et al., 2023) with 33.4M parameters for watermark extraction. The model training is performed on images from the SA-1b dataset (Kirillov et al., 2023) for 600k steps with a batch size of 256. We use AdamW optimizer with a learning rate of $5{\times}10^{-4}$, decayed via a cosine schedule, and a linear warm-up period of 20k steps. We apply our high-resolution training recipe described in Section 4.3, but for practical reasons, we limit the maximum image size to $S_{max}{\times}S_{max}$. To further ensure robustness against different image aspect ratios, we randomly resize the image into a width/height sampled from the interval $[S_{min}, S_{max}]$. We set $S_{min} = 256$ and $S_{max} = 768$. During the fist stage training we set the watermark scaling factor to $\alpha_0 = 1.0$ and decrease it during the second stage to the final value of $\alpha_1 = 0.2$. The released Pixel Seal model is trained with the discriminator of Rombach et al. (2022) with the default watermark boosting factor $\beta = 1$. The objectives are weighted with $\lambda_{\text{msg}} = 1.0$ and $\lambda_{\text{adv}} = 0.1$. We do not use any perceptual loss. Our models encode $n_{\text{bits}} = 256$ bits.

Baselines. We compare Pixel Seal with other publicly available multi-bit watermarking methods. We consider CIN (Ma et al., 2022) with 30-bit message, InvisMark (Xu et al., 2025) with 94-bit message[1], MBRS (Jia et al., 2021) with 256-bit message, TrustMark (Bui et al., 2023) with 100-bit message, Video Seal 0.0 (Fernandez et al., 2024) with 96-bit message, Video Seal 1.0 with 256-bit message, and WAM (Sander et al., 2025) with 32-bit message. For video watermarking, we also compare with RivaGAN (Zhang et al., 2019) with 32 32-bit message. Similarly to Pixel Seal, the baselines also operate at a fixed resolution, i.e., we use Equation (9) when evaluating them on high-resolution content.

Evaluation details. We evaluate the methods using 1000 images of resolution 1280×1280 generated by Meta AI using the prompts from Gowal et al. (2025) and 100 real high-resolution photos from the SA-1b validation set (Kirillov et al., 2023). For video evaluation, we further use a dataset of 121 videos[2] generated by Movie Gen (Polyak et al., 2024) and an SA-V validation set of 96 high-quality videos. The evaluation is performed on the first 3 seconds of each video. We measure both watermark imperceptibility and robustness. For imperceptibility, we use PSNR, SSIM, LPIPS (Zhang et al., 2018), CVVDP (Mantiuk et al., 2024), and our just-noticeable difference (JND) metric. The JND metric is computed using a regressor trained on internal user study data to measure the perceptual quality difference between the watermarked image and the reference. A difference of 1 JND means that 75% of observers would find the difference noticeable[3]. For evaluating watermark robustness, we apply various transformations (e.g., brightness change, cropping, compression) and report both the bit accuracy and $\log_{10} p$. The latter represents the logarithm of the probability of observing the measured bit accuracy by chance (Equation (2)). This allows for a fair comparison between methods with different payload sizes (we refer to Fernandez et al. (2024, App. A) for more information on $\log_{10} p$). The full list of evaluated transformations, along with visual examples, is available in the appendix.

5.2 Comparison with the state-of-the-art

In this section, we evaluate the robustness and imperceptibility of all image watermarking methods. Additionally, we also compare the robustness of Pixel Seal with other methods in video watermarking. For evaluation on videos, Pixel Seal uses temporal watermark pooling with step size $k = 4$ and depth $n = 2$.

Robustness. For image watermarking, the results in Table 1 for both Meta AI-generated images and real SA-1b photos show that all methods perform well against simple valuemetric attacks, such as a small brightness

[1] InvisMark encodes 100-bit message; however, due to its random message sampling process during training, 6 bits of the message have their values fixed.

[2] The videos are available at www.youtube.com/playlist?list=PL86eLlsPNfyi27GSizYjinpYxp7gEl5K8.

[3] Please note that the JND metric described here differs from the JND heatmap used to attenuate the watermark on flat areas.

change, and image compression algorithms, such as JPEG. However, for geometric attacks, such as cropping, we see Pixel Seal clearly outperforming all other methods. The performance difference is further amplified in combined attacks consisting of cropping, JPEG compression, and brightness change, where only Pixel Seal and InvisMark maintain bit accuracy above 90% while Pixel Seal embeds a significantly larger message than InvisMark (256 vs. 94 bits). In video watermarking, compression algorithms such as H.264 and HEVC present a significantly greater challenge, resulting in lower bit recovery rates across all methods, especially in the combined attack where JPEG image compression is replaced by H.264 video compression. Nonetheless, Pixel Seal still outperforms all related methods.

Imperceptibility. We evaluate watermark imperceptibility both quantitatively and qualitatively. The quantitative evaluation of the images generated by Meta AI is presented in Table 2. We can see Video Seal 0.0 substantially outperforms all other methods in the PSNR and SSIM metrics. However, this is due to its very localized artifacts, which yield, on average, a high metric value but are very visible. This can be seen in Figure 5 and is further confirmed by other quality metrics that perform more complex aggregation than just averaging. Overall, across all quality metrics, only Pixel Seal and TrustMark-P rank among the top three methods for 4 out of 5 metrics. However, TrustMark-P is significantly less robust than Pixel Seal. Strong imperceptibility of Pixel Seal watermarks is further confirmed by the qualitative results in Figure 5. Pixel Seal produces watermarks that are localized to the edges of objects in the image and are of small magnitude. In contrast, for example, WAM produces a very localized watermark, but its magnitude is much larger, or InvisMark produces watermarks of a small magnitude but not well localized. Additional qualitative results are available in the appendix.

Table 1 Robustness evaluation. We evaluate the robustness of all methods under various attacks on image (SA-1b and Meta AI images) and video datasets (MovieGen and SA-V videos) using the original content resolution. In addition to the bit accuracy, we report negative $\log_{10} p$ to account for the different number of bits encoded by each method. Pixel Seal constantly outperforms all related methods, especially in the case of difficult geometric and combined attacks. The image and video attacks differ; the full list of attacks is available in the appendix.

		Identity		Valuemetric		Compression		Geometric		Combined	
		Bit acc. (↑)	$-\log_{10} p$ (↑)	Bit acc. (↑)	$-\log_{10} p$ (↑)	Bit acc. (↑)	$-\log_{10} p$ (↑)	Bit acc. (↑)	$-\log_{10} p$ (↑)	Bit acc. (↑)	$-\log_{10} p$ (↑)
Meta AI (*images*)	CIN	1.00	9.0	0.85	7.5	1.00	9.0	0.52	0.7	0.50	0.4
	InvisMark	1.00	28.3	0.89	22.0	0.98	26.1	0.77	14.8	0.94	22.1
	MBRS	1.00	**75.9**	0.95	62.0	1.00	**75.2**	0.52	3.6	0.50	0.4
	TrustMark-P	1.00	29.6	0.89	21.2	0.98	27.1	0.65	8.2	0.52	0.7
	TrustMark-Q	1.00	30.0	0.97	27.1	1.00	29.8	0.67	9.4	0.54	0.9
	Video Seal 0.0	0.97	25.0	0.84	16.7	0.93	20.9	0.80	13.3	0.74	7.4
	Video Seal 1.0	0.98	69.4	0.95	60.7	0.95	60.6	0.83	42.9	0.69	16.0
	WAM	1.00	9.6	0.89	7.5	1.00	9.5	0.77	4.7	0.62	1.6
	Pixel Seal (ours)	1.00	75.8	0.97	**68.1**	0.98	68.9	0.95	**63.9**	0.91	**50.3**
SA-1b (*photos*)	CIN	1.00	9.0	0.85	7.5	1.00	9.0	0.52	0.7	0.50	0.4
	InvisMark	1.00	28.3	0.89	21.4	1.00	27.8	0.77	14.1	0.98	25.7
	MBRS	0.99	71.8	0.93	55.9	0.99	71.1	0.52	3.4	0.50	0.4
	TrustMark-P	0.99	28.8	0.84	17.3	0.98	26.7	0.64	7.7	0.52	0.6
	TrustMark-Q	1.00	29.9	0.96	25.9	1.00	29.7	0.65	8.5	0.53	0.8
	Video Seal 0.0	0.98	25.2	0.85	17.2	0.96	23.3	0.79	12.6	0.74	7.6
	Video Seal 1.0	0.99	72.8	0.96	64.6	0.99	71.0	0.82	41.8	0.70	17.1
	WAM	1.00	9.6	0.90	7.7	1.00	9.6	0.78	4.8	0.72	3.0
	Pixel Seal (ours)	1.00	**75.4**	0.98	**68.5**	0.99	**73.6**	0.93	**59.0**	0.94	**55.6**
MovieGen	RivaGAN	0.93	7.2	0.83	5.1	0.81	4.5	0.58	1.0	0.56	0.7
	Video Seal 0.0	0.98	26.3	0.88	20.0	0.86	16.7	0.82	14.4	0.70	6.9
	Video Seal 1.0	1.00	76.1	1.00	74.9	0.89	**51.5**	0.88	50.0	0.63	12.1
	Pixel Seal (ours)	1.00	**76.9**	1.00	**76.5**	0.86	48.6	0.97	**68.0**	0.70	**25.6**
SA-V	RivaGAN	0.97	8.2	0.87	6.1	0.82	4.7	0.60	1.3	0.56	0.7
	Video Seal 0.0	0.99	27.3	0.89	21.1	0.81	14.6	0.83	15.7	0.64	4.5
	Video Seal 1.0	1.00	76.6	1.00	76.0	0.87	**49.4**	0.87	50.4	0.59	8.0
	Pixel Seal (ours)	1.00	**77.1**	1.00	**76.8**	0.86	48.1	0.98	**70.8**	0.68	**24.2**

Table 2 Quantitative evaluation of the imperceptibility. We report the visual quality metrics on the 1000 images generated by Meta AI. Only Pixel Seal and TrustMark-P rank among the top three methods for 4 out of 5 metrics. However, TrustMark-P is significantly less robust than Pixel Seal.

	PSNR (↑)	SSIM (↑)	CVVDP (↑)	LPIPS (↓)	JND (↓)
CIN	44.3	0.9894	9.51	0.0274	1.93
InvisMark	46.6	0.9823	9.93	0.0020	0.26
MBRS	46.3	0.9928	9.54	0.0049	1.59
TrustMark-P	49.0	0.9908	**9.94**	0.0018	**0.19**
TrustMark-Q	43.3	0.9886	9.73	0.0022	1.66
Video Seal 0.0	**53.7**	**0.9989**	9.85	0.0039	1.07
Video Seal 1.0	48.7	0.9958	9.84	**0.0012**	0.45
WAM	41.4	0.9760	9.72	0.0344	0.95
Pixel Seal (ours)	48.9	0.9905	9.87	0.0013	0.34

5.3 Ablations

We ablate all key components introduced in Section 4: adversarial-only loss function with watermark boosting, three-stage training schedule, high-resolution training, and temporal watermark pooling. The main ablation results are shown in Table 3 and are detailed next.

Adversarial-only loss function. Unlike related work, our Pixel Seal model is trained using only adversarial loss (Equation (6)) and message loss (Equation (3)). To justify our choice of loss function, we also train our model using a mean-squared error perceptual loss, a common approach in related work. In Table 3 (section **(a)**), we show the results for training runs with different perceptual loss weights λ_{perc}. Training with a small $\lambda_{perc} = 0.1$ has only a small effect on the final results in terms of both watermark robustness and imperceptibility. When the perceptual loss weight is increased to $\lambda_{perc} = 1.0$, we observe that the model is no longer able to learn any watermark, yielding a random bit accuracy of 50%. We stop this experiment after 100 epochs of no change in bit accuracy and therefore do not report quality metrics.

To show the effectiveness of removing visible artifacts using the discriminator, we also train our model without the adversarial loss ($\lambda_{adv} = 0$). From the visual quality metrics in Table 3 (section **(a)**), it is clear that the discriminator guides Pixel Seal to produce much less visible visual artifacts. Similarly, the JND map limits the locations of watermark artifacts in the image to areas of high pixel variance (edges), yielding a much less visible watermark.

Watermark boosting. To control the imperceptibility of the watermark, we vary the watermark boosting parameter β. With $\beta > 1$, the watermark is artificially amplified for the discriminator, making it easier to detect and remove. In Table 3 (section **(b)**), we verify that selecting $\beta = 2.5$ significantly reduces visibility of artifacts while slightly decreasing the model's robustness. Setting $\beta = 0.5$ yields the opposite results, i.e., more visible artifacts with improved robustness.

Three-stage training schedule. To stabilize the training process, Pixel Seal is trained with a three-stage schedule that allows the model to first learn how to produce robust watermarks. Only later is the model guided to make these robust watermarks less visible. In Table 3 (section **(c)**), we show the results when the model is trained with the final watermark scaling factor $\alpha = 0.2$ from the beginning, as well as when the model is trained with the adversarial loss from the beginning. In both cases, the model is unable to learn any watermark, resulting in a random bit accuracy of 50%. We stop this experiment after 100 epochs of no change in bit accuracy and therefore do not report quality metrics.

High-resolution training. During training, we compute the JND map, apply the discriminator, and perform all attacks in high resolution instead of using the low 256×256 resolution of the watermarking models. As shown in Table 3 (section **(d)**), the high-resolution training outperforms the baseline approach in robustness for all types of attacks. Additional improvement is achieved in geometric and combined attacks due to the variable image aspect ratio. We also observe a reduction in visible watermark artifacts due to the application of the discriminator in high resolution.

Table 3 Ablations of the key Pixel Seal improvements. The results show our key contributions (adversarial-only loss function, three-stage training schedule, and high-resolution adaptation) significantly contribute to the state-of-the-art performance of Pixel Seal. A large drop in the metrics is indicated in red. The results are reported on the SA-1b validation set of 100 photos.

		Identity	Valuemetric	Compression	Geometric	Combined	PSNR (↑)	JND (↓)
		Bit acc. (↑)	Bit acc. (↑)	Bit acc. (↑)	Bit acc. (↑)	Bit acc. (↑)		
(a)	Pixel Seal	1.00	0.98	0.99	0.93	0.94	47.0	0.44
	w/ perceptual loss ($\mathcal{L}_{perc} = L_2^2$, $\lambda_{perc} = 0.1$)	0.99	0.97	0.99	0.92	0.91	47.1	0.41
	w/ perceptual loss ($\mathcal{L}_{perc} = L_2^2$, $\lambda_{perc} = 1.0$)	0.50	0.50	0.50	0.50	0.50	–	–
	w/o discriminator ($\lambda_{adv} = 0$)	1.00	0.99	1.00	0.95	0.97	44.9	0.96
	w/o JND	1.00	0.98	1.00	0.93	0.97	43.1	1.34
(b)	$\beta = 0.5$	1.00	0.98	1.00	0.95	0.96	46.2	0.56
	Pixel Seal ($\beta = 1$)	1.00	0.98	0.99	0.93	0.94	47.0	0.44
	$\beta = 2.5$	0.99	0.96	0.98	0.90	0.86	50.1	0.30
(c)	Pixel Seal	1.00	0.98	0.99	0.93	0.94	47.0	0.44
	w/o watermark scaling ($\alpha_0 = \alpha_1 = 0.2$)	0.50	0.50	0.50	0.50	0.50	–	–
	w/o discriminator delay	0.50	0.50	0.50	0.50	0.50	–	–
(d)	Pixel Seal	1.00	0.98	0.99	0.93	0.94	47.0	0.44
	fixed 256×256 resolution	0.97	0.94	0.96	0.81	0.68	46.8	0.45

Table 4 Ablation of the temporal watermark pooling parameters. Inserting the temporal average pooling layer after the first U-Net block ($d = 1$) produces a speedup of 2.2x compared to baseline Pixel Seal. The results are reported on the SA-V validation set of 96 videos.

	Identity	Valuemetric	Compression	Geometric	Combined	PSNR (↑)	Speedup (↑)
	Bit acc. (↑)	Bit acc. (↑)	Bit acc. (↑)	Bit acc. (↑)	Bit acc. (↑)		
Pixel Seal ($k = 1$)	1.00	1.00	0.86	0.98	0.68	47.6	–
w/ temporal pooling ($k = 4$, $d = 3$)	1.00	1.00	0.85	0.98	0.69	47.7	1.0x
w/ temporal pooling ($k = 4$, $d = 2$)	1.00	1.00	0.85	0.98	0.69	47.6	1.7x
w/ temporal pooling ($k = 4$, $d = 1$)	1.00	1.00	0.84	0.98	0.66	47.7	2.2x

Temporal watermark pooling. For efficient watermarking of videos using Pixel Seal, we insert a temporal average pooling layer into the embedder, reducing the size of the internal video feature representations. We measure the robustness, imperceptibility, and relative speedup of pooling with a kernel size of $k = 4$ at various depths d. In Table 4, we see that inserting the temporal average pooling layer after the first U-Net block ($d = 1$) produces a speedup of 2.2× compared to baseline, while having minimal impact on performance. The inference speed was measured on the Quadro GP100 GPU using 64 video frames and a vanilla PyTorch implementation of the model.

6 Conclusion

In this work, we present Pixel Seal, a state-of-the-art image watermarking model both in terms of robustness and imperceptibility. The imperceptibility of the generated watermarks is achieved through a novel adversarial-only approach that utilizes watermark boosting without relying on perceptual losses. Further, due to our three-stage training with high-resolution adaptation, the training process is stable even under heavy augmentations, resulting in extremely robust watermarks. Finally, we introduce an inference-time technique for adapting image watermarking models to video, achieving a substantial speedup without compromising the model's performance. We release Pixel Seal model weights to foster further research in this field.

References

Mauro Barni, Franco Bartolini, and Alessandro Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE transactions on image processing*, 2001.

Satyendra Biswas, Sunil R Das, and Emil M Petriu. An adaptive compressed mpeg-2 video watermarking scheme. *IEEE transactions on Instrumentation and Measurement*, 2005.

Adrian G Bors and Ioannis Pitas. Image watermarking using dct domain constraints. In *ICIP*, 1996.

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. https://openai.com/research/video-generation-models-as-world-simulators, 2024.

Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Universal watermarking for arbitrary resolution images. *arXiv preprint arXiv:2311.18297*, 2023.

Wes Castro. Invisible watermarking: Content provenance for videos at scale. https://engineering.fb.com/2025/11/04/video-engineering/video-invisible-watermarking-at-scale/, 2025.

Xuanming Chang, Beijing Chen, Weiping Ding, and Xin Liao. A dnn robust video watermarking method in dual-tree complex wavelet transform domain. *Journal of Information Security and Applications*, 2024.

Ching-Yeh Chen, Shao-Yi Chien, Yu-Wen Huang, Tung-Chien Chen, Tu-Chih Wang, and Liang-Gee Chen. Analysis and architecture design of variable block-size motion estimation for h.264/avc. *IEEE Transactions on Circuits and Systems*, 2006.

Luan Chen, Chengyou Wang, Xiao Zhou, and Zhiliang Qin. Robust and compatible video watermarking via spatio-temporal enhancement and multiscale pyramid attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. *arXiv preprint arXiv:2404.14055*, 2024.

Steffen Czolbe, Oswin Krause, Ingemar Cox, and Christian Igel. A loss function for generative neural networks based on watson's perceptual model. *NeurIPS*, 2020.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *ICCV*, 2023.

Pierre Fernandez, Hady Elsahar, I Zeki Yalniz, and Alexandre Mourachko. Video seal: Open and efficient video watermarking. *arXiv preprint arXiv:2412.09492*, 2024.

Google. Veo 3 technical report. https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf, 2025.

Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, et al. Synthid-image: Image watermarking at internet scale. *arXiv preprint arXiv:2510.09263*, 2025.

Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers. In *CVPR*, 2024.

Kun Hu, Zixuan Hu, Qianhui Zhu, Xiaochao Wang, and Xingjun Wang. Stegavideo: Robust high-resolution video steganography with temporal and edge guidance. 2024.

Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *ACM Multimedia*, 2021.

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *ICCV*, 2021.

Nikola Jovanović, Ismail Labiad, Tomáš Souček, Martin Vechev, and Pierre Fernandez. Watermarking autoregressive image generation. In *NeurIPS*, 2025.

Louis Kerner, Michel Meintz, Bihe Zhao, Franziska Boenisch, and Adam Dziedzic. Bitmark: Watermarking bitwise autoregressive image generative models. In *NeurIPS*, 2025.

Changhoon Kim, Kyle Min, Maitreya Patel, Sheng Cheng, and Yezhou Yang. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. *arXiv preprint arXiv:2306.04744*, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Wai-Kin Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. In *ICLR*, 2025.

Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *CVPR*, 2020.

Xiyang Luo, Yinxiao Li, Huiwen Chang, Ce Liu, Peyman Milanfar, and Feng Yang. Dvmark: a deep multiscale framework for video watermarking. *IEEE Transactions on Image Processing*, 2023.

Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *ACM Multimedia*, 2022.

Rafal K Mantiuk, Param Hanji, Maliha Ashraf, Yuta Asano, and Alexandre Chapiro. Colorvideovdp: A visual difference predictor for image, video and display distortions. *arXiv preprint arXiv:2401.11485*, 2024.

Aayush Mishra, Suraj Kumar, Aditya Nigam, and Saiful Islam. Vstegnet: Video steganography network using spatio-temporal features and micro-bottleneck. In *BMVC*, 2019.

Najla Mohaghegh and Omid Fatemi. H.264 copyright protection with motion vector watermarking. In *International Conference on Audio, Language and Image Processing*, 2008.

Nikos Nikolaidis and Ioannis Pitas. Robust image watermarking in the spatial domain. *Signal processing*, 1998.

Maneli Noorkami and Russell M Mersereau. A framework for robust watermarking of h.264-encoded video with controllable detection performance. *IEEE Transactions on information forensics and security*, 2007.

Minzhou Pan, Yi Zeng, Xue Lin, Ning Yu, Cho-Jui Hsieh, Peter Henderson, and Ruoxi Jia. Jigmark: A black-box approach for enhancing image watermarks against diffusion model edits. *arXiv preprint arXiv:2406.03720*, 2024.

Aleksandar Petrov, Pierre Fernandez, Tomáš Souček, and Hady Elsahar. We can hide more bits: The unused watermarking capacity in theory and in practice. *arXiv preprint arXiv:2510.12812*, 2025.

Alessandro Piva, Mauro Barni, Franco Bartolini, and Vito Cappellini. Dct-based watermark recovering without resorting to the uncorrupted original image. In *Proceedings of international conference on image processing*, 1997.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022.

Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. *ICLR*, 2025.

Xiaofeng Shen, Heng Yao, Shunquan Tan, and Chuan Qin. Vhnet: A video hiding network with robustness to video coding. *Journal of Information Security and Applications*, 2023.

Matthieu Urvoy, Dalila Goudia, and Florent Autrusseau. Perceptual dft watermarking with improved detection and robustness to geometrical distortions. *IEEE Transactions on Information Forensics and Security*, 2014.

Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. A digital watermark. In *Proceedings of 1st international conference on image processing*, 1994.

Bingyang Wen and Sergul Aydore. Romark: A robust watermarking system using adversarial training. *arXiv preprint arXiv:1910.01221*, 2019.

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.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *CVPR*, 2023.

Xiang-Gen Xia, Charles G Boncelet, and Gonzalo R Arce. Wavelet transform based watermark for digital images. *Optics Express*, 1998.

Rui Xu, Mengya Hu, Deren Lei, Yaxi Li, David Lowe, Alex Gorevski, Mingyu Wang, Emily Ching, and Alex Deng. Invismark: Invisible and robust watermarking for ai-generated image provenance. In *WACV*, 2025.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *CVPR*, 2024.

Guanhui Ye, Jiashi Gao, Yuchen Wang, Liyan Song, and Xuetao Wei. Itov: efficiently adapting deep learning-based image watermarking to video watermarking. In *International Conference on Culture-Oriented Science and Technology*, 2023.

Chong Yu. Attention based data hiding with generative adversarial networks. In *AAAI*, 2020.

Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry S Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. In *ICLR*, 2021.

Chaoning Zhang, Adil Karjauv, Philipp Benz, and In So Kweon. Towards robust deep hiding under non-differentiable distortions for practical blind watermarking. In *ACM Multimedia*, 2021.

Honglei Zhang, Hu Wang, Yuanzhouhan Cao, Chunhua Shen, and Yidong Li. Robust watermarking using inverse gradient attention. *arXiv preprint arXiv:2011.10850*, 2020.

Jing Zhang, Anthony TS Ho, Gang Qiu, and Pina Marziliano. Robust video watermarking of h. 264/avc. *IEEE transactions on circuits and systems*, 2007.

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Xiaohui Zhang, Weisi Lin, and Ping Xue. Just-noticeable difference estimation with pixels in images. *Journal of Visual Communication and Image Representation*, 2008.

Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *CVPR*, 2024a.

Xuanyu Zhang, Youmin Xu, Runyi Li, Jiwen Yu, Weiqi Li, Zhipei Xu, and Jian Zhang. V2a-mark: Versatile deep visual-audio watermarking for manipulation localization and copyright protection. In *ACM Multimedia*, 2024b.

Zhiwei Zhang, Han Wang, Guisong Wang, and Xinxiao Wu. Hide and track: Towards blind video watermarking network in frequency domain. *Neurocomputing*, 2024c.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, 2018.

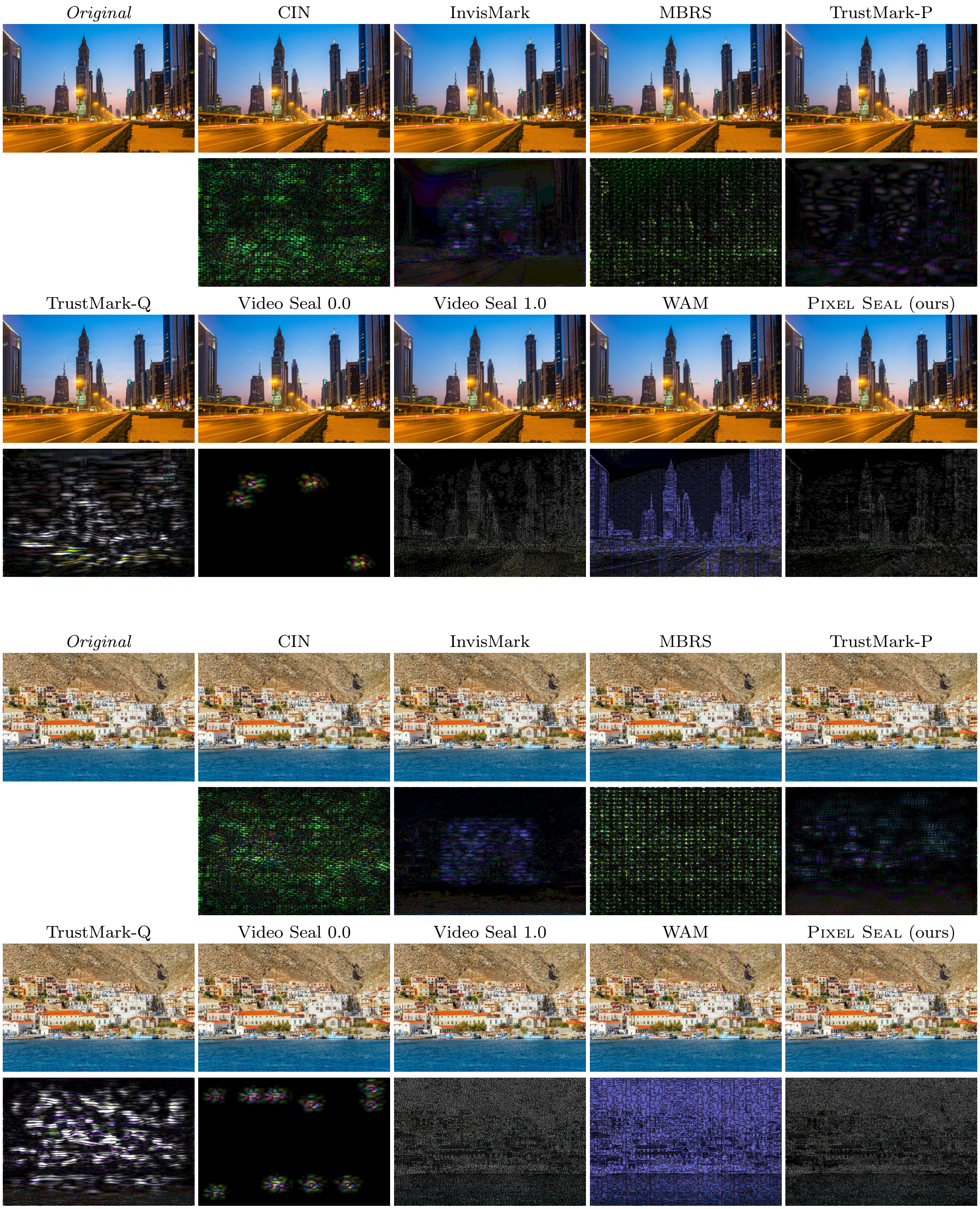

Figure 6 Comparison with related work on photos from the SA-1b dataset. We show both the watermarked image (top) and the predicted watermark brightened for clarity (bottom). Many related methods leave visible artifacts in areas with a single color. In contrast, Pixel Seal does not leave visible artifacts in such areas while being more robust to various transformations.

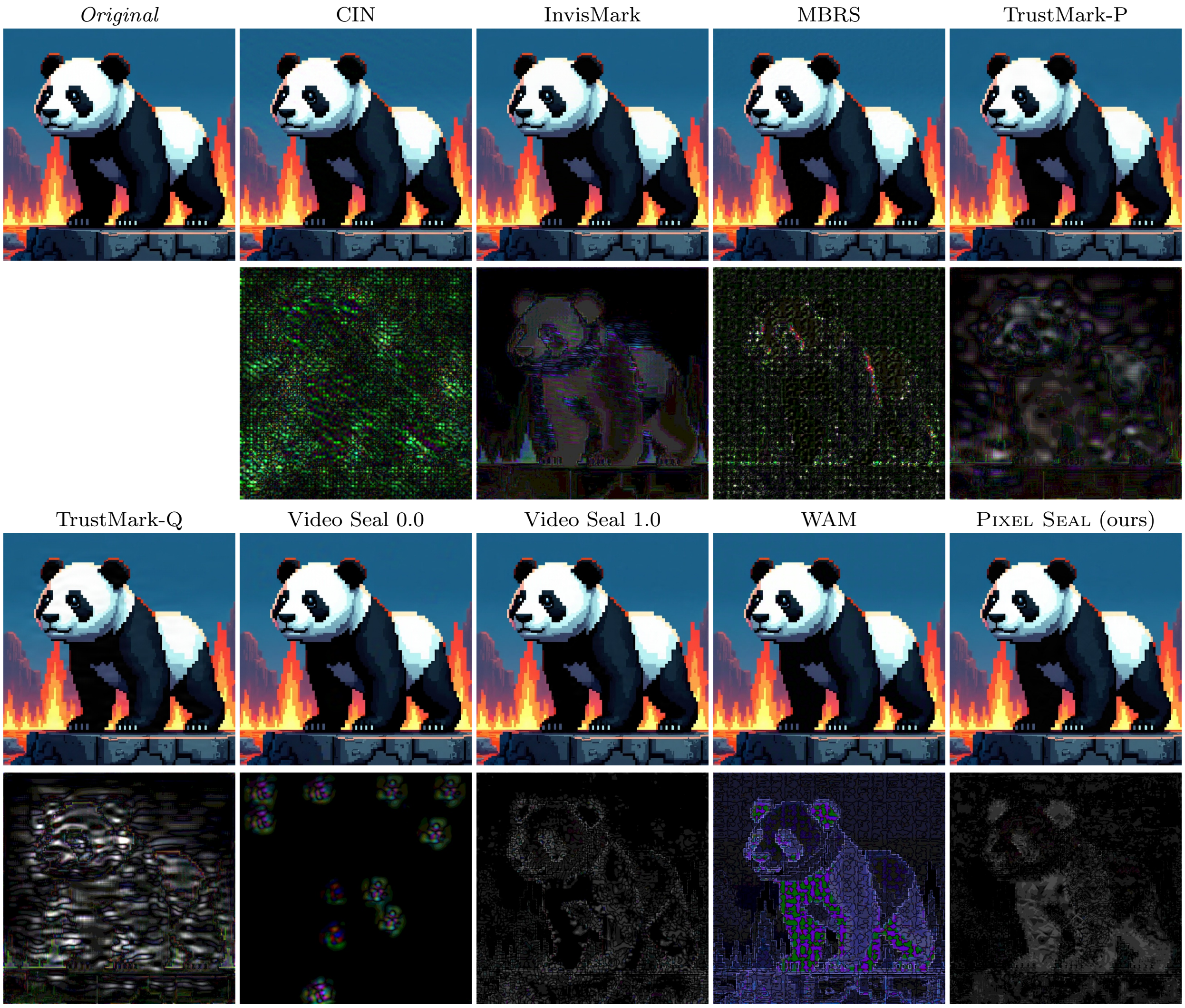

Figure 7 Comparison with related work on an AI-generated image. We show both the watermarked image (top) and the predicted watermark brightened for clarity (bottom). Many related methods leave visible artifacts in areas with a single color. In contrast, Pixel Seal does not leave visible artifacts in such areas while being more robust to various transformations.

Valuemetric attacks (images)

Brightness 0.1	Brightness 0.25	Brightness 0.5	Brightness 0.75	Brightness 1.0	Brightness 1.25	Brightness 1.5	Brightness 1.75	Brightness 2.0	Contrast 0.1	Contrast 0.25	Contrast 0.5	Contrast 0.75	Contrast 1.0	Contrast 1.25	Contrast 1.5	Contrast 1.75	Contrast 2.0	Hue -0.4	Hue -0.3	Hue -0.2	Hue -0.1	Hue 0.0	Hue 0.1	Hue 0.2	Hue 0.3	Hue 0.4	Hue 0.5	Grayscale	GaussianBlur 3	GaussianBlur 5	GaussianBlur 9	GaussianBlur 13	GaussianBlur 17
0.80	0.98	1.00	1.00	1.00	0.96	0.94	0.93	0.91	0.85	0.99	1.00	1.00	1.00	0.96	0.95	0.94	0.92	0.98	0.99	0.99	0.99	1.00	0.99	0.98	0.99	0.99	0.97	1.00	1.00	1.00	1.00	1.00	1.00

Geometric attacks (images)

HorizontalFlip	Rotate 5	Rotate 10	Rotate 30	Rotate 45	Rotate 90	Crop 0.32	Crop 0.45	Crop 0.55	Crop 0.63	Crop 0.71	Crop 0.77	Crop 0.84	Crop 0.89	Crop 0.95	Crop 1.0	Perspective 0.1	Perspective 0.2	Perspective 0.3	Perspective 0.4	Perspective 0.5	Perspective 0.6	Perspective 0.7	Perspective 0.8
1.00	0.99	0.99	0.99	0.98	0.99	0.58	0.81	0.90	0.95	0.98	0.99	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99	0.98	0.96	0.93	0.87

Compression attacks (images)

JPEG 40	JPEG 50	JPEG 60	JPEG 70	JPEG 80	JPEG 90
0.96	0.97	0.97	0.99	0.99	1.00

Combined attacks (images)

(JPEG Crop Brightness) (40 0.71 0.5)	(JPEG Crop Brightness) (60 0.71 0.5)	(JPEG Crop Brightness) (80 0.71 0.5)
0.86	0.91	0.96

Valuemetric attacks (videos)

Brightness 0.5	Brightness 1.5	Contrast 0.5	Contrast 1.5	Saturation 0.5	Saturation 1.5	Hue 0.25	Grayscale	GaussianBlur 9
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Geometric attacks (videos)

HorizontalFlip	Rotate 10	Rotate 90	Crop 0.55	Crop 0.71	Perspective 0.5
1.00	1.00	0.95	0.95	1.00	1.00

Compression attacks (videos)

JPEG 40	H264 23	H264 30	H264 40	H264 50	H264rgb 23	H264rgb 30	H264rgb 40	H264rgb 50	H265 23	H265 30	H265 40	H265 50
1.00	1.00	0.98	0.77	0.52	1.00	0.99	0.90	0.70	1.00	0.98	0.74	0.54

Combined attacks (videos)

(H264 Crop Brightness) (23 0.71 0.5)	(H264 Crop Brightness) (30 0.71 0.5)	(H264 Crop Brightness) (40 0.71 0.5)	(H264 Crop Brightness) (50 0.71 0.5)
0.95	0.76	0.51	0.50

Figure 8 The full list of attacks used for the image evaluation (top) and video evaluation (bottom). For each attack, we also report the bit accuracy of Pixel Seal on the Meta AI images (top) and SA-V videos (bottom). Selected attacks are visualized in Figure 9.

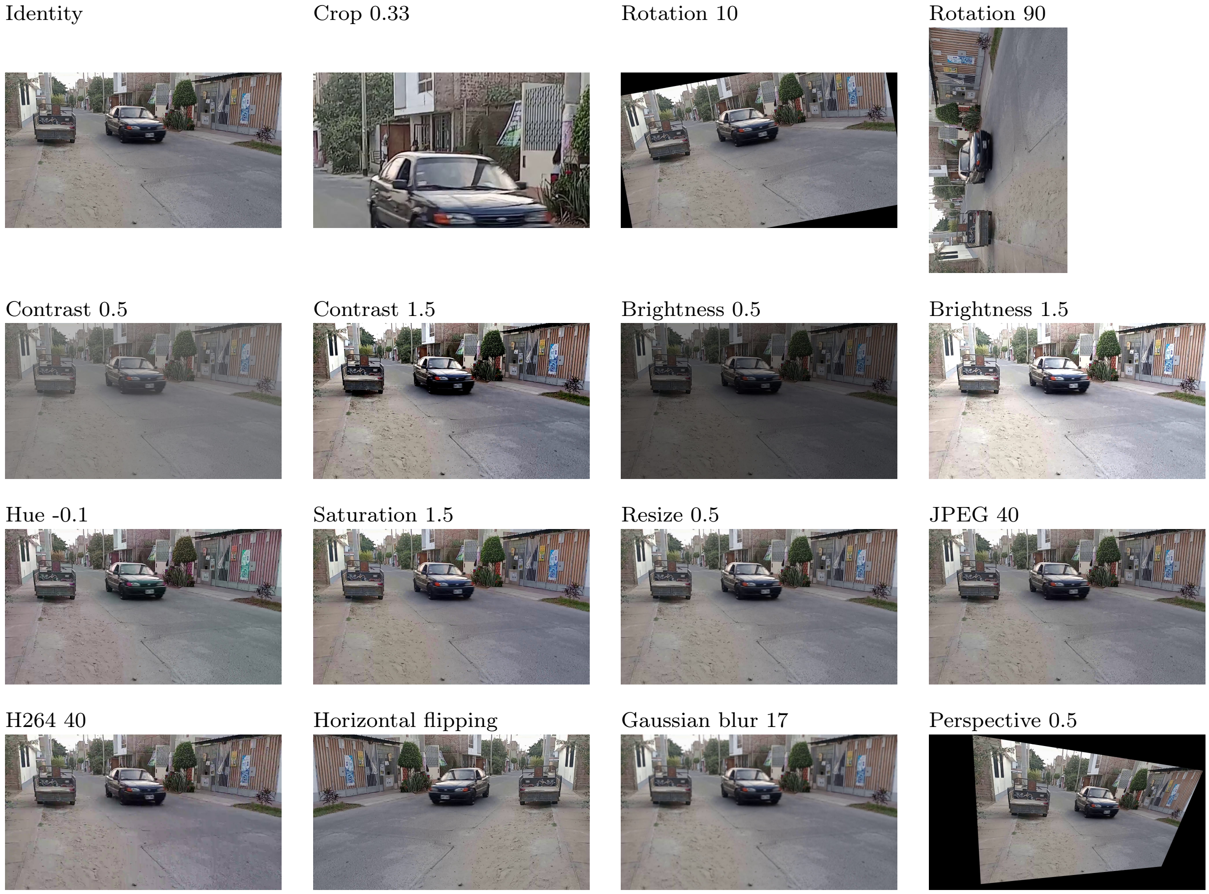

Figure 9 Illustration of some of the selected transformations.