# On Spatial-Temporal Robust Watermarking With Pseudo Random Codes

Shai Dickman, Shanaya Malik, Alex Tong, Esha Garg

December 12, 2025

**Abstract**

The introduction of generative models has ushered in an era of eerily realistic synthetic content across a broad range of media, including images, audio, and language. Although the capabilities of these models are undoubtedly impressive, the rapid adaptation raises serious concerns about misinformation and copyright abuse. As a result, it is crucial to have reliable methods for identifying artificially generated content. Generative watermarking addresses this challenge by embedding a statistical signal directly into the model's sampling process, offering superior robustness and undetectability compared to classical post-hoc methods. However, while watermarking has been extensively studied for language and image models, video generation models remain relatively underexplored, despite their growing presence on social media platforms. The temporal aspect of videos introduces new vulnerabilities—such as clipping and frame dropping—that complicate detection. In this report, we extend the PRC watermark [Gunn et al., 2024] to the video domain, specifically analyzing its robustness against temporal attacks and characterizing the fundamental trade-offs between capacity, imperceptibility, robustness, and undetectability.

## I. Introduction

To address the risks of AI-generated content, a new paradigm of watermarking has emerged: generative watermarking. Instead of embedding watermarks post-hoc, generative watermarking embeds a statistical signal directly into the video creation process, treating it as a communication channel, yielding watermarks that are inherently more robust and imperceptible than classical schemes. However, while watermarking static images has been well studied over the past few years, with relatively robust solutions against various spatial attacks, generative video watermarking remains significantly underexplored due to its novelty. Videos possess a temporal dimension, introducing a new class of adversarial attacks. In real-world pipelines, videos are frequently subjected to *temporal desynchronization*: they may be clipped to shorter durations, dropped frames during streaming (packet loss), or subsampled to lower frame rates. For a watermarking system to be practical, it must be robust to these temporal distortions without requiring the detector to have access to meta-information such as which frames were removed. In this work, we address the challenge of robust video watermarking by extending the PRC watermark [Gunn et al., 2024] to the spatio-temporal domain. Using the Open Sora 1.3 architecture as our testbed, we demonstrate that naive extensions of image-based watermarking schemes fail catastrophically under temporal attacks and that a more careful construction is needed to achieve temporal robustness.

Our contributions are as follows:

- **Spatial Robustness Baseline**: We extend the evaluation of the PRC watermark for images under geometric attacks not covered in the original work, establishing a critical baseline for understanding the synchronization sensitivity that subsequently guides our video analysis.
- **Evaluation of Natural Spatio-temporal Extension**: We formulate a natural extension of the PRC watermark from 2D spatial latents to 3D spatio-temporal latents. While we validate its imperceptibility using the VBench benchmark, we demonstrate that this naive approach is extremely brittle, failing even under minor temporal shifts due to the strict index-dependence of the 3D key.
- **Analysis of Temporal Broadcasting & Phase Mismatch**: To address this brittleness, we investigate "Temporal Broadcasting" (redundancy over time). We identify a critical "Phase Mismatch" vulnerability that causes this strategy to still fail under front-clipping attacks. We show this occurs because removing frames shifts the alignment of physical pixels relative to the model's fixed latent compression blocks (typically 4:1), scrambling the embedded signal.
- **Modulo-4 Alignment Strategy**: We propose a robust detection protocol that resolves these synchronization issues by cycling through temporal phase shifts during extraction. We show that this strategy recovers the watermark under severe attacks, maintaining detectability even with $55\%$ video removal or random frame dropping with probability up to $p = 0.75$.
- **Discovery of the "Anchor Effect"**: We characterize the fundamental operating limits of our method, explicitly identifying the "Anchor Effect" under subsampling attacks. We demonstrate that detection in subsampled videos is essentially an artifact of the model's initialization, relying strictly on the presence of the very first physical frame to preserve the watermark signal.

## II. Background

Watermarking can be viewed as a communication problem: an encoder embeds a signal into content, an adversary applies a channel transformation (e.g., compression, cropping, re-encoding), and a decoder attempts to detect or recover the message. A helpful way to organize prior work is therefore not only by algorithmic construction, but by *where* the watermark is embedded (pixel space, transform space, or model latent space) and *which distortions* it is designed to survive.

*a) Post-hoc watermarking (pixel and transform domains).:* We consider representative classical watermarking techniques that operate directly on generated samples:

- **Least Significant Bit (LSB).** LSB embeds information by modifying the lowest bit planes of pixel values. While it offers high capacity and can be imperceptible under ideal conditions, it is extremely brittle in practice. In our experiments, LSB survived essentially only lossless operations and failed under typical compression, filtering, and spatial perturbations, confirming its unsuitability for adversarial settings.
- **Discrete Fourier Transform (DFT).** DFT-based watermarking embeds information in the magnitude or phase of the frequency spectrum. Because the Fourier representation captures more global image structure than raw pixels, DFT exhibits improved robustness to certain post-processing operations and mild geometric transformations. Among traditional methods, DFT performed best in our evaluation, outperforming LSB under pixel-level distortions and mild geometric transformations such as translation and rotation; however, it still failed under stronger geometric attacks, such as cropping and rescaling.

*b) Generative watermarking (model-native embedding).:* Recent work moves watermarking upstream into the sampling process of diffusion and flow-matching models by modifying the initial Gaussian latent.

- **Tree-Ring Watermarking.** Tree-ring watermarking injects a circular band into the frequency domain of the initial latent, typically concentrating energy in higher frequencies to preserve perceptual quality. It demonstrates strong empirical robustness in image diffusion models [Wen et al., 2023].
- **Gaussian-Shading Watermarking.** Gaussian-shading biases the initial Gaussian latent toward a key-dependent region while preserving the marginal distribution of individual samples [Yang et al., 2024]. Despite its robustness, prior work reports a reduction in sample diversity, highlighting a trade-off between detectability and generative richness [Gunn et al., 2024].
- **Pseudorandom Codes (PRC).** PRC embeds a pseudorandom code into the initial latent to achieve strong detectability without sacrificing sample diversity [Gunn et al., 2024]. A recent extension applies PRC to video by distributing the code across spatiotemporal latents and then recovering code alignments with edit distance calculations [Hu et al., 2025]. Still, it evaluates only mild temporal distortions (e.g., single-frame drops). In this work, we study PRC under *severe* spatial and temporal attacks and analyze how latent compression architectures fundamentally constrain robustness.

*c) Summary and motivation.:* Classical post-hoc schemes degrade rapidly when the adversarial channel introduces strong desynchronization, whereas generative watermarking gains leverage by embedding signals at the model's natural initialization point. This distinction motivates our focus on PRC, which we evaluate across aggressive spatial and temporal attacks to characterize its operating limits and recovery strategies in both image and video generation models.

## III. Generative Model Inference & Inversion

As described earlier, it can be valuable to embed watermarks within the sampling process of a generative model rather than after generation. Most image and video generation models use iterative processes in the latent space, initialized with Gaussian noise. A common technique is to watermark the initial Gaussian latent using a detectable transformation. To extract the watermark, we need a way to predict the initial latent from a given sample. Thus, an understanding of the inner workings of these generative models is essential. We provide a brief overview of the sampling and inverse sampling methods of Stable Diffusion 1.5 [Rombach et al., 2021], a Diffusion image model, and Open Sora 1.3 [Zheng et al., 2024], a Flow-Matching video model. In the following sections, we investigate how to watermark outputs from these models by using corresponding sampling and inverse sampling algorithms.

### A. Pretrained Variational Autoencoders

A Variational Autoencoder (VAE) is a model that encodes a sample into a latent representation and then decodes the latent closely to its original state. VAE models consist of an encoder and a decoder, and pretrained VAEs are often used in generative models that operate on a latent space. The encoders are explicitly designed to compress the sample space while preserving local structure. Stable Diffusion's 2D autoencoder maintains spatial fidelity via a patch-based code-book [Esser et al., 2021, Rombach et al., 2021]. Open Sora's 3D autoencoder uses similar 2D compression as well as causal 3D CNNs to compress groups of four frames into a single latent [Yu et al., 2024, Zheng et al., 2024]. The structure preservation of encoders is relevant to understand how to inject a watermark in a latent since the structure of the injection will likely be preserved in the corresponding decoded sample.

### B. Stable Diffusion 1.5 Sampling

The iterative processes that define Diffusion models are called the forward and reverse processes. Figure 1 shows these processes in the Stable Diffusion architecture. Forward diffusion iteratively adds Gaussian noise to a data point $x_0$ to achieve

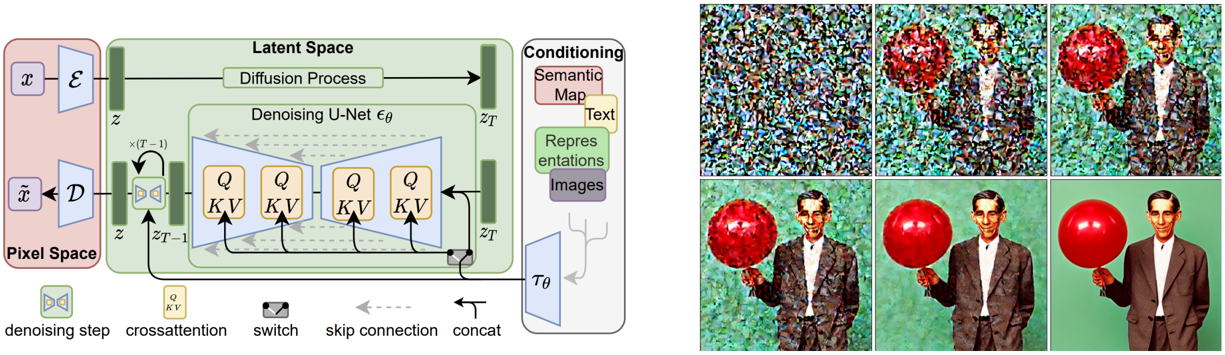

Fig. 1. Stable Diffusion model architecture [Rombach et al., 2021] (left), Stable Diffusion 1.5 example: "Claude Shannon holding a red balloon" (right).

a final noised result $x_T$. Here, $x_0$ is the latent representation of an image, which is derived by a pretrained encoder. There is also a corresponding decoder that converts latents to images for the reverse process. The forward process can be written in a closed form as

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

where $\bar{\alpha}_t$ is a function of the scheduled variances for the iteratively added noise, and the $\epsilon$ term represents the cumulative Gaussian noise. Inference is done via the reverse (or denoising) process, which generates a sample latent given an initial Gaussian latent. An example of this process, conditioned on a test prompt, is depicted in Figure 1. Denoising Diffusion Implicit Models (DDIM) is a commonly used process for sampling and is used to sample from Stable Diffusion 1.5 [Song et al., 2022, Rombach et al., 2021]. At each denoising step, we form a prediction $\epsilon_\theta$ of the cumulative noise added to $x_0$ to get $x_t$ during the forward process. We use this noise prediction to form an estimate $\hat{x}_0^t$ of the initial sample $x_0$. Finally, we predict $x_{t-1}$ using the DDIM update step:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \hat{x}_0^t + \sqrt{1-\bar{\alpha}_{t-1}} \epsilon_\theta(x_t) \tag{2}$$

We approximate the inversion of this method in the same way as the Tree-ring watermarking paper as well as several others [Wen et al., 2023]. This inversion assumes $x_t - x_{t-1} \approx x_{t+1} - x_t$. It is as follows:

$$x_{t+1} = \sqrt{\bar{\alpha}_{t+1}} \hat{x}_0^t + \sqrt{1-\bar{\alpha}_{t+1}} \epsilon_\theta(x_t) \tag{3}$$

This method is an approximation and will accumulate errors from each additional step.

### *C. Open Sora 1.3 Sampling*

Flow-Matching is another generative process with strong parallels to Diffusion models; however, it is formulated differently. The forward process is a linear deterministic interpolation between the data point $x_0$ and a noise term $\epsilon$ [Lipman et al., 2023]. In Open Sora, $x_0$ is the latent representation of an entire video. Like Stable Diffusion, Open Sora uses an encoder and decoder to map between pixel/temporal space and latent space. It also lets the noise term $\epsilon$ be Gaussian, which makes the forward process correspond to a Diffusion forward process. The key difference between the latent representation of Open Sora and Stable Diffusion is that Open Sora utilizes an additional temporal dimension. In addition, while Diffusion models are trained to predict the additive noise at a given timestep, Flow-Matching models are trained as vector fields $v_\theta$ that can be integrated via standard ODE solvers to derive samples $\hat{x}_0$ from an initial state of Gaussian noise [Lipman et al., 2023]. These two methods are very similar; in fact, the DDIM update step shares the same structure as Euler's method, a first-order accurate scheme for solving ODEs commonly used in Flow-Matching inference [Xu et al., 2025]. The Flow-Matching update step is:

$$x_{n-1} = x_n + (t_n - t_{n-1}) v_\theta(x_n; t_n) \tag{4}$$

Operating under the same assumption as DDIM inversion that $x_t - x_{t-1} \approx x_{t+1} - x_t$, we use the following inverse step:

$$x_n = x_{n-1} - (t_n - t_{n-1}) v_\theta(x_{n-1}; t_n) \tag{5}$$

Like DDIM, this inversion raises accuracy concerns because it is still an approximation. For an exact inversion, we can solve the implicit update step

$$x_n = x_{n-1} - (t_n - t_{n-1}) v_\theta(x_n; t_n) \tag{6}$$

but this is computationally more demanding as it will require multiple fixed-point iterations. This method has been used for image editing which requires higher accuracy inversion [Xu et al., 2025], however, like with DDIM, we found that an approximate inversion is sufficient for extracting the watermark.

### D. Consequences of Approximate Inversions

It is important to acknowledge an additional challenge of injecting a watermark into the initial latent of a generative model: the inverse process to predict the initial latent can, in itself, degrade the watermark, even without any attacks on the original sample. The inverse processes for Stable Diffusion and Open Sora, while slightly different, are both approximations and accumulate error. Consider a watermarked image $X$, an attacked image $Y$ and the predicted noise latent $f(Y)$. We can view the errors introduced by $f$ as simply additional image attacks. Then $X - Y - f(Y)$ form a Markov Chain. By the Data Processing Inequality, $I(X, f(Y)) \leq I(X, Y)$. It follows that the capacity of a watermark extracted from the approximate inverted latent representation $f(Y)$ is upper bounded by the capacity of a watermark extracted directly from $Y$. Thus, if $f$ is not a bijection, we will have lower watermark capacity by embedding it into the initial Gaussian latent. Even still, generative watermarking is appealing because the initial latent Gaussian distribution creates an excellent opportunity for elegant schemes with much less impact on visual quality and yet also more robustness than many standard ad-hoc methods.

## IV. PRC Watermarking

One of the most important properties of watermarks in the real world is **imperceptibility**, or the idea that a watermark should be subtle and not detectable by humans. This also includes not degrading the quality of the output. Since a large amount of money and energy is put into fine tuning these models, ensuring quality preservation is essential if watermarks are to be adopted in practice. Imperceptibility has often been calculated empirically via scores like CLIP or FID.

However, [Christ et al., 2024] proposed a new, formalized definition for a very strong imperceptibility guarantee, which they call **undetectability**. For any output with a high enough empirical entropy, undetectability states it is infeasible to distinguish between the distributions of $\overline{\mathsf{Model}}$ (A random variable representing the $\mathsf{Model}$ response) and $\mathsf{Wat}_{\mathsf{sk}}$ (The distribution over watermarked outputs, given a secret key sk), even when those can be queried adaptively with arbitrary prompts.

The definition in [Christ et al., 2024] is restated below.

*Definition 1 (Computational Undetectability):* A watermarking scheme $\mathcal{W} = (\mathsf{Setup}, \mathsf{Wat}, \mathsf{Detect})$ is *undetectable* if for every security parameter $\lambda$ and all polynomial-time distinguishers $\mathcal{D}$,

$$\left| \Pr\left[\mathcal{D}^{\mathsf{Model},\overline{\mathsf{Model}}}(1^\lambda) \to 1\right] - \Pr_{\mathsf{sk}\leftarrow\mathsf{Setup}(1^\lambda)}\left[\mathcal{D}^{\mathsf{Model},\mathsf{Wat}_{\mathsf{sk}}}(1^\lambda) \to 1\right] \right| \leq \mathsf{negl}(\lambda).$$

Here, the notation $\mathcal{D}^{\mathcal{O}_1,\mathcal{O}_2}$ means that $\mathcal{D}$ is allowed to adaptively query both oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ with arbitrary prompts.

Note that the Distinguisher is given access to both the actual $\mathsf{Model}$, as well as the distributions of watermarked and un-watermarked outputs.

[Christ and Gunn, 2024] proposes a novel approach to watermarking by replacing the randomness used by the generation process in the model with codewords from a pseudorandom error-correcting code (PRC). This means that the randomness used by the model is is replaced by a pseudorandom distribution, which is indistinguishable from random without the secret watermarking key sk. This in turn means that model outputs have an undetectable watermark, providing an extremely strong quality guarantee and ensuring that no adversary can learn the watermark.

In the next section, we discuss the definition and properties of a PRC and provide an overview of the construction in [Christ and Gunn, 2024].

### A. PRC Details

A *pseudorandom code* (PRC) [Christ and Gunn, 2024] is defined by algorithms $\mathsf{Encode}$ and $\mathsf{Decode}$ satisfying two properties:

- *Pseudorandomness:* Any efficient adversary, without knowledge of the decoding key, cannot distinguish between oracle access to $\mathsf{Encode}$ and an oracle that always outputs a freshly sampled random string.
- *Error correction (robustness):* For any message $m$, if $x \leftarrow \mathsf{Encode}(m)$ and $x'$ is a "corrupted" version of $x$ where the amount of corruption is bounded, then $\mathsf{Decode}(x') = m$.

Pseudorandomness and error-correction are properties that tend to be in direct conflict. Without the secret key, the output must look completely unstructured. With the secret key, the output must look structured enough to allow for robust error-correction. One of the main contributions of [Christ and Gunn, 2024] is bypassing this tradeoff. Here, we focus on the zero-bit case, where the watermark encodes a single bit $\mathsf{b} \in \{0, 1\}$, representing whether the watermark is present or not present in the output. The essential idea is to use a Low Density Parity Check (LDPC) code for error correction, and pair it with a parity based assumption from cryptography – Learning Parity with Noise (LPN). LDPC codes were originally introduced by Gallager and shown to be capacity achieving for different symmetric channels. In [Christ and Gunn, 2024], due to the need for pseudorandomness and undetectability via the LPN assumption, their parity check matrices employ higher density and thus necessitate different decoding techniques than the typical belief propagation for LDPC codes.

The LPN assumption states that noisy samples from the codespace of a random linear code are pseudorandom, even to an adversary who knows a generator matrix for the code. In more detail, let $n, g$ be integers and let $G \leftarrow \mathbb{F}_2^{n\times g}$ be a random matrix. The LPN assumption (with noise rate $\eta$ and secrets of size $g$) states that

$$(G, Gs \oplus e) \approx (G, u),$$

where $s \leftarrow \mathbb{F}_2^g$, $e \leftarrow \text{Ber}(n, \eta)$, and $u \leftarrow \mathbb{F}_2^n$.

However, under the LPN, the output is indistinguishable even given $G$, i.e. there is nothing we can use as a secret key to distinguish the two outputs. We need something that can *detect* when a codeword is not randomly generated. For this, we turn to LDPC codes. That is, instead of sampling $G$ uniformly at random, we first sample a *parity-check matrix* $P \in \mathbb{F}_2^{r\times n}$ subject to each row being $t$-sparse, where $r$ is the number of parity checks. Then we sample $G \in \mathbb{F}_2^{n\times g}$ subject to $PG = 0$.

For appropriate choices of $n, g, t, r$, it can be shown that the resulting marginal distribution on $G$ is random or pseudorandom [Christ and Gunn, 2024]. The pseudorandomness of $G$ will allow us to apply the LPN assumption, so that the outputs look indistinguishable from random to any adversary. The parity check matrix $P$ will allow us to efficiently detect near-codewords.

1) Sample a random matrix $P \in \mathbb{F}_2^{r\times n}$ subject to every row of $P$ being $t$-sparse.
2) Sample a random matrix $G \in \mathbb{F}_2^{n\times g}$ subject to $PG = 0$.
3) Output $(P, G)$.

The corresponding encoding function is

$$\text{Encode}_G(1): \text{ Sample } s \leftarrow \mathbb{F}_2^g \text{ and } e \leftarrow \text{Bernoulli(n}, \eta). \text{ Output } Gs \oplus e.$$

Zero-bit decoding works by counting the number of satisfied parity checks, and checking if they are over some threshold. We expect that for $u \leftarrow F_2^n$, we will have $d(Pu, 0) \approx r/2$. For a message $s$ encoded as $x = Gs \oplus e$, even after corruptions we should expect it will have small Hamming distance from Range$(G)$ so that $d(Px, 0) < r/2$ (the noise $e$ is usually set to be small).

The corresponding decoding function which bounds the number of unsatisifed parity checks is derived in [Christ and Gunn, 2024] as:

$$\text{Decode}_P(x): \ ||Px)||_0 < (1/2 - r^{-1/4}) \cdot r, \text{ output 1; otherwise output } \perp .$$

The paper determines that with constant noise rate, the sparsity of the parity check matrix must be $t = O(\log r)$ to ensure decoding succeeds with high probability (i.e bound the number of satisfied parity checks to be close to $r$). Furthermore, $r = n^{\Omega(1)}$ so $t = O(\log n)$.

LDPC codes typically utilize very small constant sparsity $t << n$ [Borwankar and Shah, 2020]. However, to ensure the pseudorandomness of $G$, the sparsity is set to $t = \Theta(\log n)$. Unfortunately, a belief propagation decoder does not work for noise rates beyond $O(\log t/t)$ according to [Christ and Gunn, 2024] which is why the simple parity check decoding is used.

In [Gunn et al., 2024], the decoding function is modified to optimize for images by using a softmax function instead of bit strings, and giving more weight to values with higher magnitude. In addition, they use a constant $t = 3$, which means that the scheme is not practically undetectable against motivated adversaries, given it is too small. However, it is still undetectable in practice, as shown by the undetectability discussion in the next section. Instead, to achieve a reasonably high $t$ that follows the $\Theta(\log n)$ required for cryptographic security, [Gunn et al., 2024] discusses that a value around $t = 7$ should be used. Still both the PRC paper and our experiments use $t = 3$ which is practically sufficient.

Now that we have described the general PRC structure, we explain how it was proposed for injection in the initial Gaussian latent of a generative model. The signs of a Gaussian are randomly distributed in $\{+1, -1\}$ so we can simply replace the signs of the initial latent with the bits in the code. In particular, given a PRC $c \in \{0, 1\}^n$ and an initial Gaussian latent $x \sim \mathcal{N}(0, I_n)$, we define the new Gaussian latent as

$$\hat{x}_i = (-1)^{(c_i)} |x_i|$$

Importantly, because of the pseudorandomness of $c$, we can assume $\hat{x} \sim \mathcal{N}(0, I_n)$.

### *B. Robustness of PRC*

- **Pixel-level Attacks**: The PRC paper shows that the watermark is robust to especially pixel-level attacks [Gunn et al., 2024]. These attacks include photometric distortions such as contrast adjustments and degradation distortions such as noise and compression. For these attacks, PRC performed similarly to the Tree-ring watermark and slightly worse than the Gaussian shading watermark.
- **Regeneration Attacks**: Regeneration attacks are a family of attacks which use the diffusion process to naturally remove the watermark. This method alters an image's latent representation using the generative model architecture [Zhao et al., 2024]. The PRC paper implemented two attacks of this kind: diffusion model-based and VAE-based attacks. The diffusion attacks add Gaussian noise to the latent of the final sample and then execute additional denoising steps. The VAE attacks use pretrained autoencoders to compress the images. PRC performs worse than Tree-ring and Gaussian shading for these evaluations. Importantly, these attacks retain the semantic quality of the original images and are particularly harmful to pixel-level watermarks [Zhao et al., 2024]. The PRC is embedded directly in the latent without any additional transformation, unlike Tree-ring, so the PRC paper's claim that "the watermark operates at a semantic level" may not be entirely true. Its vulnerability to these attacks as well as the structure preservation of VAE architectures, as will be

discussed later in this report, indicate that PRC is embedded more closely to the pixel-level and thus has areas for future improvement to be embedded in the semantic-level.
- **Undetectability**: An immense strength of PRC is its undetectability. To establish this empirically, the PRC paper trained a neural network model to detect a watermark without the key [Gunn et al., 2024]. The model was trained on techniques including Tree-ring, Gaussian Shading, and PRC. PRC was the only technique that the model could not learn to detect the watermark. Thus, while Gaussian Shading is generally more robust to the previously described attacks, it decreases sample diversity significantly and is in fact more detectable than PRC. This is likely because all images generated via Gaussian Shading correspond to a similar Gaussian quadrant which is learnable whereas the PRC, within a limited adversarial compute budget, still functions as a random Gaussian.

## V. Experiments

### A. Experiment Setup

Our experimental design is divided into two phases: spatial robustness and temporal robustness. The first phase establishes a baseline by evaluating the PRC watermark's robustness against spatial desynchronization attacks on images generated by Stable Diffusion 1.5. The second phase, which forms the core of this work, extends the PRC watermark to the temporal domain using Open Sora 1.3 and rigorously evaluates its resilience against video-specific desynchronization attacks.

To build intuition for the video domain, we first examine the analogous effects of spatial desynchronization—specifically, cropping and rotation—on watermarked images generated by the Stable Diffusion 1.5 text-to-image model. While these geometric attacks were not discussed in the original PRC paper [Gunn et al., 2024], they serve as critical benchmarks in the broader watermarking literature [Wen et al., 2023] and are a necessary precursor to our analysis of video.

### B. Stable Diffusion 1.5

While PRC was initially proposed and evaluated without geometric attacks [Gunn et al., 2024], spatial perturbations such as cropping and rotation are standard robustness benchmarks in classical and generative watermarking [Wen et al., 2023]. These attacks serve as direct spatial analogs to temporal clipping and frame dropping in videos.

*a) Attack Setup:* We evaluate masked cropping and rotation attacks that simulate everyday post-processing operations. For cropping, removed regions are replaced with black pixels, preserving the original image dimensions and isolating the effect of spatial deletion without introducing rescaling artifacts.

| **Spatial Attack** | **Decodable Limit** | **Detectable Limit** |
|---|---|---|
| Center crop (masked) | $< 30\%$ removed | $< 50\%$ removed |
| Corner crop (masked) | $< 25\%$ removed | $< 40\%$ removed |
| Rotation ($90°$, $270°$) | *All fail* | |

"Removed" denotes the percentage of total image area occluded.

TABLE I

**Spatial Robustness Limits for Images (Stable Diffusion 1.5).** Empirical breakdown points for PRC under spatial desynchronization.

*b) Discussion:* The different levels of cropping robustness exemplify the PRC's error-correction capability. However, the PRC fails for rotations. As we will discuss later, this is a result of spatial restructuring, which causes latent restructuring and thus a broken code. These image-based failure modes closely parallel the temporal desynchronization effects observed in the video domain, motivating our subsequent analysis of PRC under temporal attacks in Open Sora 1.3.

### C. Open Sora 1.3

We extend the PRC watermark to the video domain using Open Sora 1.3. We approached this problem through three iterative implementations, moving from a naive theoretical extension to a more practical solution.

Crucially, Open Sora 1.3 operates on a compressed spatiotemporal latent space. The model compresses temporal data such that one latent frame corresponds to 4 real video frames. Consequently, our watermarking key $K$ exists in the dimensions $(W \times H \times T_{latent})$, while the attacks occur on the generated video in the pixel domain $(W_{pixel} \times H_{pixel} \times T_{real})$.

*1) Attempt 1: The Native Spatiotemporal Extension:* To start, we directly extended the PRC embedding scheme from the 2D spatial latent space $(W \times H)$ used in image models to the whole 3D spatiotemporal latent space $(W \times H \times T_{latent})$. To do this, we generate a unique watermarking key $K \in \mathbb{R}^{W \times H \times T_{latent}}$ that spans the entire volume of the generated video.

**Imperceptibility Analysis.** Before evaluating robustness, however, we first established the imperceptibility of the watermark by evaluating it on VBench. This standard video generation benchmark has been shown to align well with human preferences [Zheng et al., 2025]. This benchmark comprises 2000 curated image prompts targeted at 16 metrics, including quality, semantic, and total scores. For each prompt, we generate 5 watermarked and unwatermarked 2-second video clips at 360p.

| Model | Total (%) | Quality (%) | Semantic (%) |
|---|---|---|---|
| Open-Sora 1.3 | **78.43** | **81.57** | 65.89 |
| Open-Sora 1.3 w/ PRC | 78.32 | 81.40 | **66.04** |

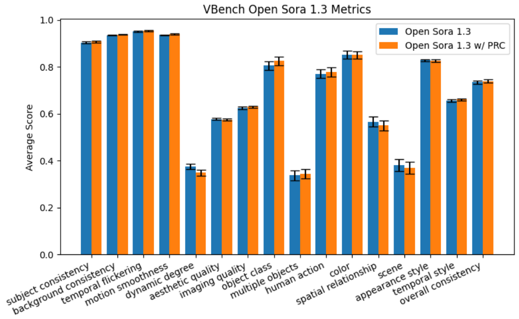

Fig. 2. Weighted VBench summary statistics (left), Full VBench metric comparison with standard error bars (right).

The results in Figure 2 show that the PRC watermark did not significantly impact the generative ability of the Open Sora model. We conclude that, as in the original PRC paper, the distributions of the watermarked and unwatermarked Open-Sora samples are reasonably similar.

**Robustness Failure.** While the visual quality was maintained, the temporal robustness proved to be a critical failure point. Because the watermark is embedded in the latent space, but attacks occur in the real frame space, even minor temporal disruptions are catastrophic. For example, dropping a single real frame shifts the entire sequence, misaligning the subsequent groups of 4 real frames with their corresponding latent key slice $K_t$. The results in Table III show that if we pad frames exactly where they were removed, thus keeping the original latent alignment, the scheme is still robust. This is comparable to a pixel-level attack on an image. If we don't know where frames were removed and instead pad arbitrarily, then the latent structure is completely reordered, just like an image rotation. Unsurprisingly, padding in incorrect positions makes the PRC undecodable.

*2) Attempt 2: Temporal Broadcasting:* To address this fragility, we adopted a "Temporal Broadcasting" strategy. Instead of a unique key for every latent frame, we generate a single spatial watermark $k \in \mathbb{R}^{W \times H}$ and repeat it $T_{latent}$ times, creating a spatiotemporally redundant signal where every latent frame carries an identical watermark.

In theory, this redundancy should render the watermark robust to any temporal cropping; since every frame carries the same watermark $k$, the detector should not require a specific temporal index to recover the signal. However, in practice, this approach fails significantly due to the decoupling of the attack domain (pixels) and the embedding domain (latents).

More specifically, the core issue lies in the 4:1 temporal compression ratio of the Open Sora 1.3 autoencoder. When the watermark is embedded, it is placed via latent frames, each of which is a compressed representation of 4 specific physical frames (e.g., Latent 0 represents Frames 0-3). Temporal attacks, such as clipping or frame dropping, operate on the physical frames. If an attack removes a number of frames that is not a multiple of 4 (e.g., removing the first frame), the alignment of the pixel groupings shifts. For instance, if we remove the first physical frame, the sequence [Frame 1, 2, 3, 4] is encoded instead, producing a latent vector that is entirely distinct from the original watermarked latent vector generated from [Frame 0, 1, 2, 3]. Consequently, naive padding (e.g., appending black frames to the end) fails to correct this alignment, rendering the watermark undetectable under front-clipping or random frame-dropping attacks.

Additionally, we observed that subsampling attacks (e.g., retaining every 4th frame) led to near-total collapse in detection. This is expected behavior: since the Open Sora 1.3 encoder requires 4 consecutive physical frames to construct a single valid latent representation, feeding it disjoint frames (e.g., 0, 4, 8, 12) deprives it of 75% of the required temporal information. The resulting latent vectors are severely distorted, rendering the watermark signal unreadable.

Interestingly, we found that the first latent frame remained detectable even under severe subsampling. This suggests that while temporal aliasing destroys the watermark signal in subsequent latents, the initial latent frame serves as a sequence anchor and retains sufficient spatial fidelity to enable detection. Our finding further supports this hypothesis that removing just the first physical frame (Frame 0) before subsampling eliminates detection. This indicates that the watermark's survival is not merely a property of the first latent block, but is strictly contingent on the presence of the very first physical frame, likely due to the autoencoder's initialization.

*3) Attempt 2.5: Modulo-4 Alignment Strategy:* The failure of Attempt 2 (summarized in Table III) clarified that simply repeating the key is insufficient; the phase of the real-to-latent compression also matters. Because Open Sora 1.3 encodes videos at a 4:1 ratio of real to latent frames, a temporal shift/crop that is not a multiple of 4 disrupts the watermark's latent representation.

To solve this, we implemented the **Modulo-4 Alignment Strategy**. Instead of a single padding configuration, our detector evaluates four distinct temporal alignments. We shift the incoming real video frames by an offset $i \in \{0, 1, 2, 3\}$ before encoding them back into the latent space (and padding the remainder). By cycling through these four phase shifts, we ensure that at least one configuration correctly aligns the real frames to the latent used during generation.

This refined strategy successfully mitigates the synchronization errors observed in previous attempts. As shown in Table III, Attempt 2.5 achieves broad robustness across all attack vectors where Attempt 2 failed. Recall, however, that temporal robustness required preserving the very first frame of the sample video or a batch of 4 correctly ordered physical frames corresponding to 1 latent frame. We further quantify the precise operating limits of our method in Table II, demonstrating that the watermark remains reliably detectable even when over $55\%$ of the video content is removed or when frames are dropped with a probability of $p = 0.75$.

TABLE II
**Operating Limits**

| | Limits | |
|---|---|---|
| **Attack Type** | **Decodable** | **Detectable** |
| **Boundary Clip** | $< 35\%$ | $< 55\%$ |
| **Frame Drop** | $p < 0.75$ | $p < 0.75$ |
| **Subsampling** | $5\times^{*}$ | $5\times^{*}$ |

[*] See Table III for caveats.

TABLE III
**Robustness Study**

| | Clipping | | Temporal Sampling | |
|---|---|---|---|---|
| **Method** | **End** | **Front** | **Dropping** | **Subsampling** |
| **Attempt 1: Native 3D** | | | | |
| *Pad End* | ✓ | ✗ | ✗ | ✗ |
| *Pad Front* | ✗ | ✓ | ✗ | ✗ |
| **Attempt 2: Redundancy** | | | | |
| *Pad End* | ✓ | $\sim^{\dagger}$ | ✓ | $\sim^{*}$ |
| **Attempt 2.5** | | | | |
| **Modulo-4 Align** | ✓ | ✓ | ✓ | $\sim^{*}$ |

✓= Robust, ✗= Fail, $\sim$ = Conditional Success
[†] Phase Mismatch: Only detectable if $L \equiv T \pmod 4$
[*] Anchor Effect: Detection confined to first latent frame

## VI. Discussion

Our results demonstrate that spatial attacks on images, such as rotation and cropping, act as direct analogs to temporal attacks on videos. These shared failure modes stem from the VAE architectures employed by both model types. Because these architectures map spatial/temporal regions directly to corresponding latent regions, they lack the translation invariance required for PRC robustness. A rotation in pixel space or a frame drop in time completely offsets the latent grid, converting a simple synchronization shift into a catastrophic "deletion attack" on the watermark code. As established in prior work, PRC is robust to substitution but fails under deletion, as a shift at the beginning of the sequence destroys the alignment of the entire pseudorandom key [Gunn et al., 2024].

## VII. Future Directions

**Exploiting the Anchor Frame Effect.** Our discovery that watermark detection under subsampling persists strictly in the first latent frame suggests that video VAEs heavily prioritize sequence initialization. Future work should analyze the internal workings to determine if this bias can be systematically exploited.

**Generalization to Emerging Architectures.** The synchronization challenges identified in Open Sora 1.3 likely extend to any latent video model with temporal compression. Future work should evaluate these techniques on state-of-the-art architectures like Wan2.1 to determine how causal latent structures affect the difficulty of embedding robust watermarks across frames.

**Expanding Capacity via Temporal Modulation.** While temporal broadcasting maximizes robustness through redundancy, it limits capacity. We believe that the temporal dimension can be used to achieve higher watermarking capacity without sacrificing on robustness. A potential scheme for achieving this is to inject the PRC into the latent frequency domain. This approach could utilize the full temporal volume while maintaining invariance to the shift-based distortions identified in this work.

## VIII. Conclusion

As generative video models continue to advance, the ability to verify authenticity and identify AI-generated content is becoming increasingly critical. In this work, we extended the PRC watermark to the video domain, exposing a fundamental vulnerability in naive implementations: the synchronization mismatch between pixel-domain attacks and latent-domain compression. By introducing the Modulo-4 Alignment Strategy, we improved robustness to severe temporal attacks despite this vulnerability. While this protocol significantly improves detection under desynchronization, it fails to leverage the higher capacity afforded by video latents. More work is also needed to address the broader range of temporal distortions encountered in real-world pipelines and to develop watermarks that are inherently robust to these adversarial shifts.

# References

Saumya Borwankar and Dhruv Shah. Low density parity check code (ldpc codes) overview, 2020. URL https://arxiv.org/abs/2009.08645.

Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part VI*, page 325–347, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-68390-9. doi: 10.1007/978-3-031-68391-6_10. URL https://doi.org/10.1007/978-3-031-68391-6_10.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1125–1139. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/christ24a.html.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. URL https://arxiv.org/abs/2012.09841.

Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. *arXiv preprint arXiv:2410.07369*, 2024.

Xuming Hu, Hanqian Li, Jungang Li, Yu Huang, Shuliang Liu, Qi Zheng, Junhao Chen, and Aiwei Liu. Videomark: A distortion-free robust watermarking framework for video diffusion models, 2025. URL https://arxiv.org/abs/2504.16359.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust, 2023. URL https://arxiv.org/abs/2305.20030.

Pengcheng Xu, Boyuan Jiang, Xiaobin Hu, Donghao Luo, Qingdong He, Jiangning Zhang, Chengjie Wang, Yunsheng Wu, Charles Ling, and Boyu Wang. Unveil inversion and invariance in flow transformer for versatile image editing, 2025. URL https://arxiv.org/abs/2411.15843.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models, 2024. URL https://arxiv.org/abs/2404.04956.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024. URL https://arxiv.org/abs/2310.05737.

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, 2024. URL https://arxiv.org/abs/2306.01953.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.