The Coding Limits of Robust Watermarking for Generative Models

Danilo Francati*
Sapienza University of Rome
francati@di.uniroma1.it

Yevin Nikhel Goonatilake*
George Mason University
ygoonati@gmu.edu

Shubham Vivek Pawar*
Royal Holloway, University of London
shubham.pawar.2022@live.rhul.ac.uk

Daniele Venturi
Sapienza University of Rome
venturi@di.uniroma1.it

Giuseppe Ateniese
George Mason University
ateniese@gmu.edu

Abstract

We study a basic question about cryptographic watermarking for generative models: how reliable can a watermark remain when an adversary is allowed to corrupt the encoded signal? To address this question, we introduce a minimal coding abstraction that we call a *zero-bit tamper-detection code*. This is a secret-key procedure that samples a pseudorandom codeword and, given a candidate word, decides whether it should be treated as unmarked content or as the result of tampering with a valid codeword. It captures the two core requirements of robust watermarking: *soundness* and *tamper detection*.

Within this abstraction we prove a sharp unconditional limit on robustness to independent symbol corruption. For an alphabet of size q, there is a critical corruption rate of $1 - 1/q$ such that no scheme with soundness, even relaxed to allow a fixed constant false positive probability on random content, can reliably detect tampering once an adversary can change more than this fraction of symbols. In particular, in the binary case no cryptographic watermark can remain robust if more than half of the encoded bits are modified. We also show that this threshold is tight by giving simple information-theoretic constructions that achieve soundness and tamper detection for all strictly smaller corruption rates.

We then test experimentally whether this limit appears in practice by looking at the recent watermarking for images of Gunn, Zhao, and Song (ICLR 2025). We show that a simple crop and resize operation reliably flipped about half of the latent signs and consistently prevented belief-propagation decoding from recovering the codeword, erasing the watermark while leaving the image visually intact.

Keywords: watermarking, pseudorandom codes, impossibility, generative models, diffusion models.

*Equal contribution.

1 Introduction

In recent years, generative AI models such as GPT, Llama, and Claude have made it increasingly difficult to tell human-produced content from machine-generated text, code, and images. This new reality raises a central question for both technology and society: how can we reliably distinguish what is AI-generated from what is authentically human?

Watermarking is one of the most promising strategies for meeting this challenge [Exe23, Eur24]. The idea is to embed a secret, imperceptible pattern into model outputs at generation time, enabling later verification by anyone with the appropriate key. Ideally, a watermark should be robust, surviving even substantial adversarial editing, while also being undetectable to those without the key and leaving the original content unchanged in quality or meaning [Aar22, KGW+23, CG24].

Recent work by Christ and Gunn [CG24] formalized cryptographic watermarking using pseudorandom error-correcting codes (PRCs), showing that such watermarks can be both undetectable and robust to significant amounts of tampering. But their results naturally lead to a deeper question: within such cryptographically grounded, coding-based frameworks, what are the true limits of robustness? Is there an unconditional barrier that no watermarking scheme of this form can overcome?

This question has been explored from two sides. On one side, the "Watermarks in the Sand" (WiTS) work [ZEF+23, ZEF+24] showed that any sufficiently robust watermark is, in principle, vulnerable: if an adversary has enough power, especially access to quality and perturbation oracles, it can eventually erase any watermark without degrading content quality. However, WiTS leaves open the concrete thresholds for robustness that efficient watermarking schemes can actually achieve. On the other side, PRC-based schemes demonstrate the power of current cryptographic techniques but do not establish whether their level of robustness is the best possible in the coding sense.

In this paper, we bring these perspectives together. We introduce a simple abstraction, the *zero-bit tamper-detection code*, which we use as a minimal model for cryptographic watermarking based on pseudorandom codes. This notion can be viewed as the zero-bit specialization of a PRC equipped with an explicit tamper-detection predicate, and it captures the main coding issues that arise in such watermarking schemes. Using this notion, we prove a tight unconditional threshold on robustness to independent symbol corruption in this model: no scheme realizing a zero-bit tamper-detection code can reliably survive if more than half of the encoded bits are altered in the binary case. More generally, for a q-ary alphabet the corresponding limit is $(1 - 1/q)$ of the encoded symbols. Conversely, for any small constant δ, we give explicit constructions that approach these limits, providing robustness up to just under half of the bits (for binary) or just under $(1 - 1/q)$ (for q-ary) of the symbols changed.

To make our results concrete, we analyze a recent state-of-the-art PRC-based watermarking scheme for images by Gunn, Zhao, and Song [GZS25]. We show that a simple crop-and-resize operation, one that visually preserves the image but changes about half of its latent bits, is sufficient to consistently prevent belief-propagation decoding from recovering the codeword. This suggests that the theoretical limit we establish is relevant in at least one practical watermarking pipeline.

This attack also highlights a contrast between modalities: in text, watermark removal typically requires extensive edits that alter the content, while for images, even a single benign transformation can remove the watermark without changing the image in an obvious way. This makes robust watermarking in images particularly fragile, since the watermark can disappear without a visible cost.

Our findings position the impossibility results of WiTS in a precise, quantitative framework

for PRC-style cryptographic watermarks: WiTS demonstrates that all robust watermarks are eventually breakable in a powerful black-box oracle model, while our work identifies a concrete numerical threshold at which robustness becomes unconditionally impossible for schemes that can be modeled as zero-bit tamper-detection codes under independent symbol corruption.

Looking forward, our results suggest that any significant increase in robustness within this PRC-based, cryptographic framework will require fundamentally new ideas, potentially leveraging semantic or structural features of content rather than cryptographic pseudorandomness alone.

1.1 Our Contributions

Our contributions are as follows:

1. **A simplified abstraction for PRC-style watermarking.** We introduce and formalize *zero-bit tamper-detection codes*, a secret-key primitive that we use as a minimal model for cryptographic watermarking based on pseudorandom codes. These codes do not encode explicit messages but instead focus on two properties: *soundness* and *tamper detection*. They can be viewed as the zero-bit specialization of pseudorandom error-correcting codes equipped with an explicit tamper-detection predicate (and without pseudorandom codewords). We show how any watermarking scheme that is sound and robust for a given family of tampering functions induces a zero-bit tamper-detection code with the same guarantees, even when the watermark detector depends on the prompt. In this sense, zero-bit tamper-detection codes capture the core coding-theoretic requirements of PRC-based watermarking for generative models.

2. **Tight unconditional limits for watermarking schemes.** Within this abstraction we establish a sharp limit on robustness to independent symbol tampering. For an alphabet of size q there is a critical corruption rate

$$\alpha^* = 1 - \frac{1}{q}$$

   such that no zero-bit tamper-detection code can simultaneously satisfy soundness and tamper detection for independent tampering at rates exceeding α^* by any fixed constant slack. In particular, in the binary case ($q = 2$) no such code can remain both sound and robust once more than half of the encoded bits are modified. Conversely, we give a simple information-theoretic construction that achieves soundness and tamper detection for all corruption rates strictly below α^*, up to an arbitrarily small constant slack.

   We further show that this barrier persists even if soundness is weakened to *constant-bounded soundness*. For any constant $c \in (0, 1]$, once an adversary can corrupt more than an $(\alpha^* + \varepsilon)$ fraction of symbols (for any fixed $\varepsilon > 0$), any code that is sound on all but a c fraction of random strings must incur tamper-detection error close to $1 - c$ at that corruption level. In particular, even very weak schemes that behave like a coin flip on random inputs (for example, $c = 1/2$) cannot robustly detect tampering beyond this threshold.

3. **Implications for PRC-based watermarking.** By relating zero-bit tamper-detection codes to pseudorandom error-correcting codes, and using the reduction from watermarking schemes to tamper-detection codes, we obtain direct consequences for PRC-based watermarking. In particular, for the independent symbol tampering model considered by Christ and Gunn [CG24], our lower bound shows that their robustness guarantees are optimal: no cryptographic watermark (binary or q-ary, secret-key or public-key) whose robustness can

be expressed through such a coding layer can tolerate a larger fraction of symbol errors in this model.

4. **A concrete attack illustrating the limit.** Finally, we show that the coding-theoretic limit is numerically relevant for at least one state-of-the-art watermarking pipeline. We analyze the PRC-based image watermark for Stable Diffusion proposed by Gunn, Zhao, and Song [GZS25], and we study the effect of a simple crop-and-resize transformation. This transformation preserves the visual appearance of the image yet induces latent sign errors concentrated near fifty percent and consistently causes belief-propagation decoding to fail, thereby erasing the watermark.

   Earlier work had already established that watermarks can, in principle, be removed. WiTS shows this using oracle access and iterative regeneration [ZEF+23, ZEF+24], while UnMarker demonstrates removal in a fully black-box setting using optimization-based spectral attacks that slightly degrade the image [KH25]. Our case study differs in that it reaches essentially the same corruption regime using only a single, routine image operation, with no oracle access and no optimization. In this sense, it provides a concrete example where the unconditional robustness threshold arises from an ordinary edit rather than a highly tailored attack.

1.2 Related work

Watermarking refers to the process of embedding a signal in generated content such as images [BOD95, CMB+07, HD17, RP97, ZKJFF18], text [ARC+01, ARH+03], audio [Arn00, BTH96], or video that is imperceptible to humans but algorithmically detectable. Its primary use is to enable attribution or provenance without significantly degrading content quality. The study of watermarking spans decades and includes both practical systems and formal analyses, with results on constructing watermarking schemes as well as breaking or proving impossibility. With the rise of modern generative models, watermarking has become a sharper test of the trade-offs between imperceptibility, robustness, and efficiency.

Constructions. Watermarking methods are commonly classified based on when the watermark is applied: either after generation (post-processing) or during the generation process itself (in-processing). Post-processing schemes embed the watermark into the output after it has been generated. These approaches often degrade output quality [CMB+07, WWZ+22, ADR+24]. In-processing schemes, by contrast, embed the watermark during content generation. They were first introduced in image generators [YSAF21], and later adapted to large language models (LLMs) [KGW+23]. Image watermarking techniques include modifying weights through pre-training or fine-tuning [FXTB22, FCJ+23, ZZXP23, ZPD+23], or intervening in the sampling trajectory of diffusion models [WKGG23]. In the LLM setting, most methods operate at inference time by biasing token selection in a way that encodes a hidden message [KGW+23, CGZ24]. Some schemes further ensure that the overall distribution over outputs remains statistically indistinguishable from unmarked outputs [KTHL24], preserving quality.

A more recent line of work leverages cryptographic and coding-theoretic tools to design undetectable watermarking schemes. One direction [CG24] proceeds via pseudorandom error-correcting codes (PRCs) as the watermark signals. These codes can be constructed using low-density parity-check (LDPC) codes [CG24, AAC+25, GG24] and allow for robust detection under edits. PRC-based watermarking was recently applied to images [GZS25]. The main limitation [GZS25] is that although the underlying PRC is robust to modifications on a bounded number of bits, the resulting PRC-based watermarking is robust only to independent changes, limiting its applicability in real-world scenarios. In parallel, a recent public watermarking

scheme uses digital signatures, allowing third parties to verify the watermark and ensuring robustness as long as a small window of the generated content remains unmodified [FGJ+25].

Attacks and Impossibilities. Alongside construction efforts, recent work has explored the limitations of existing watermarking schemes. In text, the most prominent attack is paraphrasing [CGLS25, KSK+23, ZEF+23, ZEF+24], rewriting the watermarked content while preserving its meaning, using language models, translation tools, or manual edits. These attacks often succeed partially, but at the cost of degraded output quality. In the vision domain, a range of attacks aim to remove watermarks by adding noise to the image or latent representation, or by using optimization techniques to remove the watermark [LDFK24, SSR+24, ZZS+24]. Many of these attacks have limitations. For example, Zhao et al. [ZZS+24] focus on post-hoc watermarking methods, while Saberi et al. [SSR+24] require either white-box access to the detector or access to many watermarked and unwatermarked examples.

A more general impossibility result by Zhang et al. [ZEF+24, ZEF+23] shows that constructing robust watermarking schemes is impossible under certain conditions. They model generated outputs as nodes in a graph and prove that if an adversary has access to two oracles (a quality oracle that tells the adversary whether the output is still valid, and a perturbation oracle that allows small edits preserving semantics) then the adversary can walk through the graph until the watermark is erased without degrading content quality. This result does not specify an exact condition for when removal occurs, only that eventually such a walk succeeds.

In contrast, our result makes no oracle assumptions and provides a concrete lower bound in an independent symbol corruption model. We show that if more than half of the encoded bits are effectively randomized, then no watermark can remain both sound and robust. Prior work demonstrates that robust watermarking cannot be sustained indefinitely. Our result identifies a precise corruption threshold in this setting beyond which robustness is unconditionally impossible.

2 Preliminaries

2.1 Notation

Small letters (such as x) denote individual objects or values; calligraphic letters (such as $\mathcal{X}$) denote sets; sans-serif letters (such as A) denote algorithms. For a string $x \in \{0,1\}^*$, $|x|$ denotes its length. For a set $\mathcal{X}$, $|\mathcal{X}|$ is its cardinality. If x is chosen uniformly at random from $\mathcal{X}$, we write $x \leftarrow\!\!\$\, \mathcal{X}$. The Hamming distance $\mathrm{dist}(x, x')$ between two strings x and x' (over an alphabet Σ) is the number of positions where they differ.

For a deterministic algorithm A, $y = \mathsf{A}(x)$ means that y is the output of A on input x. For a randomized algorithm A, we write $y \leftarrow\!\!\$\, \mathsf{A}(x)$ to indicate y is sampled from the output distribution of A on input x. Alternatively, $y = \mathsf{A}(x; r)$ denotes the output when A runs on x with randomness r. An algorithm A is called *probabilistic polynomial-time* (PPT) if it is randomized and always halts in time polynomial in $|x|$.

We use $\mathsf{negl}(\lambda)$ for an arbitrary negligible function of the security parameter $\lambda \in \mathbb{N}$, i.e., for all $c > 0$, $\mathsf{negl}(\lambda) = o(\lambda^{-c})$. We use $\mathsf{poly}(\lambda)$ to denote a polynomial function of λ. Unless noted otherwise, all algorithms receive the security parameter 1^λ as input.

2.2 Watermarking for Generative Models

We recall the formalization of watermarking for generative models, following [ZEF+23, ZEF+24].

A generative model is any (possibly randomized) algorithm that, given a prompt x (such as a question or a text description), produces an output y (such as text or an image).

Definition 1 (Generative models)**.** A conditional generative model $\mathsf{Model} : \mathcal{X} \to \mathcal{Y}$ is a probabilistic polynomial-time (PPT) algorithm that, given a prompt $x \in \mathcal{X}$, outputs $y \in \mathcal{Y}$. Here, $\mathcal{X}$ is the *prompt space*, and $\mathcal{Y}$ is the *output space*. We write $y \leftarrow\!\!\$\, \mathsf{Model}(x)$ to denote sampling a response from the model on prompt x.

A watermarking scheme is a systematic way to mark the outputs of a generative model, so that later a verifier with the appropriate key can detect whether a given output is watermarked. We focus on *secret-key* watermarking: both embedding and detection require a secret key.

A watermarking scheme $\Pi = (\mathsf{Watermark}, \mathsf{Detect})$ for a family of generative models $\mathcal{M} = \{\mathsf{Model}\}$ consists of two efficient algorithms:

$\mathsf{Watermark}(1^\lambda, \mathsf{Model})$**:** On input a security parameter 1^λ and a model Model, the algorithm outputs a secret key $\kappa \in \mathcal{K}$ and a watermarked model $\mathsf{Model}_\kappa : \mathcal{X} \to \mathcal{Y}$.

$\mathsf{Detect}(\kappa, x, y)$**:** On input a secret key κ, a prompt $x \in \mathcal{X}$, and an output $y \in \mathcal{Y}$, this deterministic algorithm returns either true or false, indicating whether y is watermarked (in response to x).

A robust watermarking scheme should satisfy three key properties. First, **correctness**: watermarked outputs should always be recognized as such.

Definition 2 (Correctness of watermarking)**.** Let $\mathcal{M} = \{\mathsf{Model} : \mathcal{X} \to \mathcal{Y}\}$ be a class of generative models. We say that Π satisfies *correctness* if for every $\mathsf{Model} \in \mathcal{M}$ and for every $x \in \mathcal{X}$, we have

$$\mathbb{P}\Big[\mathsf{Detect}(\kappa, x, y) = \mathtt{true} : (\kappa, \mathsf{Model}_\kappa) \leftarrow\!\!\$\, \mathsf{Watermark}(1^\lambda, \mathsf{Model}),\ y \leftarrow\!\!\$\, \mathsf{Model}_\kappa(x)\Big] = 1.$$

Second, **soundness**: non-watermarked outputs (such as human-written or independently generated model outputs) should almost never be mistakenly detected as watermarked.

Definition 3 (Soundness of watermarking)**.** Let $\mathcal{M} = \{\mathsf{Model} : \mathcal{X} \to \mathcal{Y}\}$ be a class of generative models. We say that Π satisfies *soundness* if for every $\mathsf{Model} \in \mathcal{M}$, every $x \in \mathcal{X}$, and every fixed $y \in \mathcal{Y}$, $\mathbb{P}\big[\mathsf{Detect}(\kappa, x, y) = \mathtt{true} : (\kappa, \mathsf{Model}_\kappa) \leftarrow\!\!\$\, \mathsf{Watermark}(1^\lambda, \mathsf{Model})\big] \le \mathsf{negl}(\lambda)$.

Third, **robustness**: the watermark should survive small modifications to the output. Robustness requires that a watermark remain detectable even if an adversary modifies a small part of the output. To capture this, we consider length-preserving *tampering functions* $f : \Sigma^* \to \Sigma^*$, where $\alpha \in (0, 1)$ specifies the target fraction of edits relative to the input length.

We study two models of adversarial tampering: (1) In the *arbitrary tampering* model, the adversary may choose any positions and values for up to an α fraction of changes, allowing fully coordinated edits. (2) In the *independent tampering* model, the adversary changes each symbol independently, and we require that the total number of changes is at most an α fraction of the input length with overwhelming probability.

Let $\hat{\mathcal{F}}_\alpha$ denote the set of (possibly randomized) functions f as above (arbitrary), and let $\hat{\mathcal{F}}_\alpha^{\mathrm{ind}} \subseteq \hat{\mathcal{F}}_\alpha$ denote those which act independently on each symbol.

Definition 4 ($\hat{\mathcal{F}}$-Robustness of watermarking)**.** Let $\mathcal{M} = \{\mathsf{Model} : \mathcal{X} \to \Sigma^*\}$ be a class of generative models. A watermarking scheme Π is *$\hat{\mathcal{F}}$-robust* if for every $\mathsf{Model} \in \mathcal{M}$, every $x \in \mathcal{X}$, and every $f \in \hat{\mathcal{F}}_\alpha$, we have

$$\mathbb{P}\left[\mathsf{Detect}(\kappa, x, \tilde{y}) = \mathtt{false} \wedge \tilde{y} \neq y : \begin{array}{c}(\kappa, \mathsf{Model}_\kappa) \leftarrow\!\!\$\, \mathsf{Watermark}(1^\lambda, \mathsf{Model}) \\ y \leftarrow\!\!\$\, \mathsf{Model}_\kappa(x);\ \tilde{y} \leftarrow\!\!\$\, f(y)\end{array}\right] \le \mathsf{negl}(\lambda).$$

In our results, we primarily consider small constants $\alpha > 0$. When we refer to $\hat{\mathcal{F}}_{\alpha}^{\mathrm{ind}}$, the impossibility results are strongest, in the sense that we show robustness is already impossible against this restricted class of independent, uncoordinated edits. This immediately implies impossibility against the larger class of arbitrary tampering functions.

We always require $|f(y)| = |y|$, meaning all tampering functions are length-preserving. This restriction is standard in coding theory and only strengthens our impossibility results: if robust watermarking is impossible even when the adversary must preserve output length, it is certainly impossible when more general modifications, such as insertions or deletions, are permitted.

Our model further allows the watermarking algorithm to access the generative model's internals, and allows the detector to depend on both the prompt and output. Impossibility under these permissive conditions immediately implies impossibility in any more restricted setting.

3 Zero-bit Tamper-detection Codes

Our lower bounds will be phrased in terms of a simple coding primitive that captures the behavior of watermark detectors. Informally, we imagine that for each secret key there is a single special codeword in Σ^n. Given a string $y \in \Sigma^n$, the decoder must decide whether

1. y is the honest codeword produced under the key,
2. y is unrelated to any codeword for that key, or
3. y should be treated as a tampered version of the honest codeword.

There is no explicit message to recover; the output is only this three-way classification. We refer to such objects as *zero-bit tamper-detection codes*.

We work in a secret-key setting: both the encoder and the decoder have access to the same key sk, which is hidden from the adversary. This is a favorable model for the designer of the code. Any public-key or keyless detector can be viewed as a special case in which the key is fixed and revealed in advance. Thus, if we show that no scheme exists even in this more permissive secret-key model, the impossibility automatically applies to all more restricted variants.

It is helpful to compare this notion with a familiar primitive. Suppose we have a secure message authentication code with tagging algorithm $\mathsf{Tag}_{sk}(\cdot)$. Fix a message $m = 0$, and for a given key sk let $\gamma = \mathsf{Tag}_{sk}(0)$. We can define an encoder that always outputs γ, and a decoder that declares `valid` on input γ and `tampered` on every other input. This construction detects any change to γ perfectly. In this sense, detecting modifications of a single designated string is easy. The difficulty arises once we also require the decoder to behave well on *all* other strings that might be presented to it. The definitions below make this requirement precise.

Formally, a zero-bit tamper-detection code is specified by a triple of polynomial-time algorithms

$$\Gamma = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$$

over an alphabet Σ and a codeword space Σ^n:

$\mathsf{KGen}(1^\lambda)$**:** On input the security parameter λ, the randomized key-generation algorithm outputs a secret key sk.

$\mathsf{Enc}(sk)$**:** On input sk, the (possibly randomized) encoding algorithm outputs a codeword $\gamma \in \Sigma^n$.

$\mathsf{Dec}(sk, \gamma)$**:** On input the secret key sk and a string $\gamma \in \Sigma^n$, the deterministic decoding algorithm outputs a symbol in $\{\texttt{valid}, \texttt{invalid}, \texttt{tampered}\}$.

The first requirement is that honestly generated codewords are always accepted.

Definition 5 (Correctness of zero-bit tamper-detection codes)**.** A zero-bit tamper-detection code Γ satisfies *correctness* if, for every $\lambda \in \mathbb{N}$ and every secret key $sk \in \mathrm{supp}(\mathsf{KGen}(1^\lambda))$, we have

$$\mathbb{P}\big[\mathsf{Dec}(sk, \mathsf{Enc}(sk)) = \mathtt{valid}\big] = 1,$$

where the probability is taken over the internal randomness of Enc (if any). In particular, if Enc is deterministic then $\mathsf{Dec}(sk, \mathsf{Enc}(sk)) = \mathtt{valid}$ holds for all such sk.

To capture security, we add two further conditions. The first is *soundness*. Here we fix an arbitrary string $\hat{\gamma} \in \Sigma^n$ that is chosen independently of the key. Soundness requires that, over the random choice of sk, the decoder almost never classifies $\hat{\gamma}$ as anything other than $\mathtt{invalid}$. Intuitively, for a random key the set of strings that the decoder considers valid or tampered should occupy only a small fraction of the ambient space.

Definition 6 (Soundness of zero-bit tamper-detection codes)**.** We say that a zero-bit tamper-detection code Γ satisfies *soundness* if for every fixed string $\hat{\gamma} \in \Sigma^n$,

$$\mathbb{P}\Big[\mathsf{Dec}(sk, \hat{\gamma}) \neq \mathtt{invalid} : sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)\Big] \leq \mathsf{negl}(\lambda).$$

For our lower bounds we will also use a relaxed form of this requirement. Rather than insisting that the misclassification probability be negligible, we allow it to be bounded by a fixed constant c. From the point of view of applications, such a guarantee is very weak. For example, if $c = 1/2$, then in the worst case the decoder can do no better than a fair coin flip on a fixed string. The relaxation is useful for the structure of our result: the class of schemes that satisfy negligible soundness is contained inside the larger class that satisfy constant-bounded soundness for some $c < 1$. If we can rule out robust tamper detection for the larger class, then we have automatically ruled it out for the smaller class as well.

Definition 7 (Constant-bounded soundness of zero-bit tamper-detection codes)**.** Let $c \in (0, 1]$. We say that a zero-bit tamper-detection code Γ satisfies *constant-bounded soundness* (with parameter c) if for every fixed string $\hat{\gamma} \in \Sigma^n$,

$$\mathbb{P}\Big[\mathsf{Dec}(sk, \hat{\gamma}) \neq \mathtt{invalid} : sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)\Big] \leq c.$$

If $c = \frac{1}{2}$, we say that Γ satisfies *coin-flipping soundness*, in the sense that checking the validity of a worst-case string is no better than flipping a fair coin.

The second security condition concerns the effect of tampering on an honestly generated codeword. We model tampering by a family $\mathcal{F} = \{f : \Sigma^n \to \Sigma^n\}$ of (possibly randomized) functions. For a given key, the encoder produces $\gamma \leftarrow\!\!\$\ \mathsf{Enc}(sk)$ and the adversary applies some $f \in \mathcal{F}$ to obtain $\tilde{\gamma}$. Tamper detection requires that, except with negligible probability, the decoder outputs $\mathtt{tampered}$ whenever $\tilde{\gamma}$ differs from γ.

Definition 8 ($\mathcal{F}$-tamper detection of zero-bit tamper-detection codes)**.** We say a zero-bit tamper-detection code Γ satisfies *$\mathcal{F}$-tamper detection* if for every $f \in \mathcal{F}$,

$$\mathbb{P}\Big[\mathsf{Dec}(sk, \tilde{\gamma}) \neq \mathtt{tampered} \wedge \tilde{\gamma} \neq \gamma : sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda);\ \gamma \leftarrow\!\!\$\ \mathsf{Enc}(sk);\ \tilde{\gamma} \leftarrow\!\!\$\ f(\gamma)\Big] \leq \mathsf{negl}(\lambda).$$

Correctness, soundness, and $\mathcal{F}$-tamper detection together describe the behavior we will require from the coding layer underlying cryptographic watermarking.

3.1 Relation to PRCs

Christ and Gunn [CG24] formalize cryptographic watermarking in terms of pseudorandom error-correcting codes (PRCs). A PRC is a keyed coding scheme that both corrects a bounded amount of noise and hides its codewords from anyone who does not know the key. Since our lower bounds are meant to apply to PRC-based watermarking, it is natural to ask why we did not state them directly in that language.

The main reason is that we work in the information theoretic setting to prove the negative result.[1] It depends only on how many symbols an adversary can change and on how the decoder responds to corrupted strings. The pseudorandomness of the codeword, which is central to undetectability, plays no role in this counting argument. Zero-bit tamper-detection codes isolate exactly the pieces we need: a keyed encoder, a decoder that distinguishes `valid`, `invalid`, and `tampered`, and quantitative guarantees for soundness and tamper detection with respect to a family $\mathcal{F}$ of tampering functions. We do not constrain the distribution of codewords beyond what is needed for these properties.

Viewed this way, zero-bit tamper-detection codes form a weaker abstraction than PRCs. A zero-bit PRC must in addition make its codewords pseudorandom and is usually required to have negligible decoding error for every message. Any impossibility theorem that already rules out robust zero-bit tamper detection under these minimal assumptions therefore applies a fortiori to zero-bit PRCs that satisfy stronger guarantees. Working with the weaker interface keeps the statement of the lower bound focused on the two properties that matter for our application (soundness and tamper detection) while still covering the PRC constructions used in cryptographic watermarking.

At the same time, in the zero-bit setting the coding content of the two formalisms is essentially the same once we impose standard soundness. There is only a single message μ to recover. Given a zero-bit tamper-detection code $\Gamma = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ that satisfies soundness in the sense of Definition 6 and $\mathcal{F}$-tamper detection in the sense of Definition 8, we can view it as a (non-pseudorandom) zero-bit error-correcting code for the fixed message μ by using Enc as the encoder and defining the decoder to output μ on any input that is not classified as `invalid`. Conversely, for the purposes of our lower bound, any zero-bit PRC with negligible decoding error in a given tampering model induces a degenerate tamper-detection code by outputting `tampered` on successful decodings and `invalid` otherwise. This induced code need not satisfy correctness, since it may never output `valid`, but it preserves the soundness and tamper-detection guarantees relevant to the lower bound up to the usual negligible losses.

Thus, if we strip away pseudorandomness, zero-bit tamper-detection codes and zero-bit PRCs express the same coding-theoretic limitations in the regime we study. Our lower bound for zero-bit tamper-detection codes immediately carries over to zero-bit PRCs whose decoder is required to be robust against the same independent symbol tampering model. Adding pseudorandomness on top can improve undetectability of the watermark, but it cannot move the underlying coding limit.

Finally, the limit we obtain is tight for this abstraction. In Appendix A we describe a simple information-theoretic construction of a zero-bit tamper-detection code that tolerates any independent tampering rate strictly below the threshold identified by our bound. This matching construction shows that the impossibility is not an artifact of the definition: it captures the exact corruption level at which robust decoding, and hence robust PRC-based watermarking, becomes information-theoretically impossible.

[1] Despite this, the negative result can be naturally extended even to computational security.

3.2 Impossibility of High Tampering Rates

A central question for zero-bit tamper-detection codes is how robust they can be to tampering. If an adversary is allowed to modify a fraction of the symbols in a codeword, can the decoder still reliably detect that tampering has occurred while maintaining soundness on arbitrary strings? In this subsection we show that these two requirements come into fundamental conflict once the tampering rate is high enough.

To build some intuition, consider an extreme case in which the tampering family contains all constant functions: for every fixed string $\hat{\gamma} \in \Sigma^n$, there is a function $f_{\hat{\gamma}}$ that ignores its input and always outputs $\hat{\gamma}$. Soundness says that, over the choice of the key, the decoder should almost always label $\hat{\gamma}$ as `invalid`. On the other hand, if $\hat{\gamma}$ can arise as $f_{\hat{\gamma}}(\gamma)$ for an honestly generated codeword γ, tamper detection would like the decoder to label $\hat{\gamma}$ as `tampered`. Since the decoder can assign only one label to each string, it cannot satisfy both requirements for the same $\hat{\gamma}$. This already suggests that strong tamper detection is hard to reconcile with soundness once we allow very powerful tampering.

Our main result shows that a similar tension appears even for a much weaker, randomized tampering model in which each symbol is modified independently. Let Σ be an alphabet of size $q > 1$, and let n denote the codeword length. We focus on the following independent tampering rule: for each input $\gamma \in \Sigma^n$, the adversary produces $\tilde{\gamma}$ by choosing each symbol of $\tilde{\gamma}$ independently and uniformly from Σ, conditional on γ. For each position, the new symbol equals the original one with probability $1/q$ and differs from it with probability:

$$1 - \frac{1}{q}.$$

Call this randomized tampering function f_{unif}. Thus the expected fraction of changed positions is $1 - 1/q$. For any fixed $\delta \in (0, 1)$, a Chernoff bound gives

$$\begin{aligned} \mathbb{P}\Big[\operatorname{dist}(\gamma, f_{\text{unif}}(\gamma)) > \Big(1 - \frac{1}{q}\Big)(1+\delta)\, n\Big] \\ \leq \exp\Big(-\frac{\delta^2}{3} n\Big(1 - \frac{1}{q}\Big)\Big) \end{aligned} \tag{1}$$

for every fixed $\gamma \in \Sigma^n$. For the theorem below, it suffices that $\mathcal{F}^{\text{ind}}_{\alpha n}$ be any tampering family that contains f_{unif}, where $\alpha = \Big(1 - \frac{1}{q}\Big)(1+\delta)$; according to Equation (1), $f_{\text{unif}} \in \mathcal{F}^{\text{ind}}_{\alpha n}$ holds with overwhelming probability when considering sufficiently long codewords (i.e., large values of n).

For simplicity, we state the main impossibility theorem below under the assumption that $f_{\text{unif}} \in \mathcal{F}^{\text{ind}}_{\alpha n}$. Thus, it holds except with the probability defined in Equation (1).

Theorem 1. *Let Γ be any zero-bit tamper-detection code over Σ with codeword length $n \geq 1$ and $|\Sigma| = q > 1$. Suppose Γ satisfies $\mathcal{F}^{\text{ind}}_{\alpha n}$-tamper detection for some tampering family $\mathcal{F}^{\text{ind}}_{\alpha n}$ that contains f_{unif}, where*

$$\alpha = \Big(1 - \frac{1}{q}\Big)(1+\delta)$$

with $\delta \in (0, 1)$. Then Γ cannot satisfy soundness.

Proof. Suppose, for contradiction, that such a code Γ exists.

Let $sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$ and let $\gamma = \mathsf{Enc}(sk)$ be an honestly generated codeword. Let $\gamma' \leftarrow\!\!\$\ f_{\text{unif}}(\gamma)$.

By definition of $f_{\text{unif}} \in \mathcal{F}^{\text{ind}}_{\alpha n}$, the random variable γ' is distributed exactly as a uniformly random string in Σ^n.

By the soundness property, for every fixed $\hat{\gamma} \in \Sigma^n$,

$$\mathbb{P}[\mathsf{Dec}(sk, \hat{\gamma}) \neq \texttt{invalid}] \leq \mathsf{negl}(\lambda),$$

where the probability is over $sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$. In particular, this bound also holds when $\hat{\gamma}$ is chosen uniformly at random, so for a randomly chosen $\hat{\gamma} \leftarrow\!\!\$\ \Sigma^n$ we still have $\mathbb{P}[\mathsf{Dec}(sk, \hat{\gamma}) \neq \texttt{invalid}] \leq \mathsf{negl}(\lambda)$.

Moreover, since $f_{\text{unif}} \in \mathcal{F}_{\alpha n}^{\text{ind}}$, tamper detection gives

$$\mathbb{P}\big[\mathsf{Dec}(sk, \gamma') \neq \texttt{tampered} \wedge \gamma' \neq \gamma\big] \leq \mathsf{negl}(\lambda),$$

where the probability is over $sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$, $\gamma \leftarrow\!\!\$\ \mathsf{Enc}(sk)$, and the randomness of f_{unif}. Since γ' is uniform over Σ^n, we have $\mathbb{P}[\gamma' = \gamma] = q^{-n}$. Therefore,

$$\mathbb{P}\big[\mathsf{Dec}(sk, \gamma') = \texttt{tampered}\big] \geq 1 - \mathsf{negl}(\lambda) - q^{-n}.$$

Therefore, under the joint distribution of (sk, γ') with $sk \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$ and $\gamma' \leftarrow\!\!\$\ \Sigma^n$, we have both

$$\mathbb{P}\big[\mathsf{Dec}(sk, \gamma') = \texttt{invalid}\big] \geq 1 - \mathsf{negl}(\lambda),$$
$$\mathbb{P}\big[\mathsf{Dec}(sk, \gamma') = \texttt{tampered}\big] \geq 1 - \mathsf{negl}(\lambda) - q^{-n},$$

These events are disjoint. Since $q^{-n} \leq 1/2$ for every $q > 1$ and $n \geq 1$, these two lower bounds sum to more than 1 for all sufficiently large λ, which is impossible.

We conclude that no zero-bit tamper-detection code can be sound and $\mathcal{F}_{\alpha n}^{\text{ind}}$-tamper-detecting at tampering rates $\alpha \geq (1 - 1/q)(1 + \delta)$. □

In words, once we ask for tamper detection against independent symbol modifications that, with high probability, corrupt slightly more than a $(1 - 1/q)$ fraction of the positions, there is no way to keep the decoder sound on arbitrary fixed strings. The value

$$\alpha^* = 1 - \frac{1}{q}$$

is the critical corruption rate in this model: below it, robust tamper detection is achievable (as we show with a matching construction), while above it, soundness and tamper detection cannot coexist.

In the binary case the statement takes a particularly clean form.

Corollary 1. *Let Γ be any zero-bit tamper-detection code with alphabet $\{0,1\}$ and codeword space $\{0,1\}^n$ for $n \geq 1$. If Γ satisfies $\mathcal{F}_{\alpha n}^{\text{ind}}$-tamper detection for some tampering family $\mathcal{F}_{\alpha n}^{\text{ind}}$ that contains f_{unif}, and for*

$$\alpha = \frac{1 + \delta}{2}$$

for some $\delta \in (0,1)$, then Γ cannot also satisfy soundness.

Notice that these impossibility results do not rely on correctness: nowhere in the proof do we use the requirement that an honestly generated codeword must be accepted as `valid`. Even a degenerate scheme that never outputs `valid` would trivially satisfy soundness (since all random strings are labeled `invalid`) and might still claim to detect tampering. The theorem shows that, at high tampering rates, such a scheme also cannot satisfy the $\mathcal{F}_{\alpha n}^{\text{ind}}$-tamper detection property. This is why correctness is not assumed in our lower bounds.

Extension to weak soundness and variable length The same phenomenon persists even if we weaken soundness substantially. Recall the definition of constant-bounded soundness (Definition 7): for a parameter $c \in (0, 1]$, the decoder is allowed to misclassify a fixed string as anything other than `invalid` with probability up to c over the choice of the key. For the independent tampering model above and corruption rates

$$\alpha = \Big(1 - \frac{1}{q}\Big)(1 + \delta),$$

one can show that if a code has constant-bounded soundness with parameter c, then for large n its tamper-detection error under such tampering must be at least

$$(1 - c)\Big(1 - \exp\big(-\tfrac{\delta^2}{3}(1 - 1/q)\, n\big)\Big),$$

which tends to $1 - c$ as n grows. In particular, even a "coin-flip" decoder with $c = 1/2$ cannot hope to detect high-rate independent tampering with small error; beyond the threshold α^* it must fail with probability close to $1/2$ on tampered codewords.

Finally, the impossibility does not depend on fixing the blocklength in advance. Allowing the code to use different lengths n on different inputs does not help: for each length n we can define the corresponding independent tampering rule on Σ^n, and the same argument goes through. In this sense the barrier identified by Theorem 1 is intrinsic to the combination of soundness and strong tamper detection, rather than an artifact of a particular choice of codeword length.

4 Connecting Watermarking and Tamper-detection Codes

The fundamental limits we have established for zero-bit tamper-detection codes also govern the robustness of watermarking schemes for generative models. To make this connection precise, we show how any watermarking scheme that is sound and robust against tampering can be used to build a secret-key code with exactly the same security guarantees.

The intuition is straightforward: if a generative model can reliably embed a watermark that survives tampering, then by fixing a prompt and interpreting the model's outputs as codewords, we obtain a code that is robust to exactly the same set of manipulations. The code's decoder simply runs the watermark detector: if it finds a watermark, it signals tampering; otherwise, it outputs invalid.

For the formal theorem below, we restrict to models whose outputs have fixed length n. In that case, the resulting code fits the fixed-length zero-bit tamper-detection interface from Section 3.

Theorem 2. *Assume there exists a watermarking scheme Π for a class of generative models $\mathcal{M} = \{\mathsf{Model} : \mathcal{X} \to \Sigma^n\}$ satisfying soundness and $\hat{\mathcal{F}}$-robustness (for some family $\hat{\mathcal{F}}$).[2] Fix any model $\mathsf{Model} \in \mathcal{M}$ and any prompt $\overline{x} \in \mathcal{X}$. Then, there exists a zero-bit tamper-detection code Γ with alphabet space Σ and codeword space Σ^n satisfying soundness and $\hat{\mathcal{F}}$-tamper detection.*

Proof. We fix $\mathsf{Model} \in \mathcal{M}$ and $\overline{x} \in \mathcal{X}$ as in the theorem statement, and describe the resulting zero-bit tamper-detection code Γ.

Key generation $\Gamma.\mathsf{KGen}(1^\lambda)$**:** The key-generation algorithm outputs $sk = (\kappa, \mathsf{Model}_\kappa)$ where $(\kappa, \mathsf{Model}_\kappa) \leftarrow\!\!\$\ \Pi.\mathsf{Watermark}(1^\lambda, \mathsf{Model})$.

[2]For the sake of clarity, we assume the prompt and output space of Model is defined over the same alphabet. The result generalizes to different alphabets as well.

Encoding $\Gamma.\mathsf{Enc}(sk)$**:** The encoding algorithm outputs $\gamma = y \in \Sigma^n$ where $y \leftarrow\!\!\$\ \mathsf{Model}_\kappa(\overline{x})$.

Decoding $\Gamma.\mathsf{Dec}(sk, \gamma)$**:** The decoding algorithm lets $\gamma = y$ and runs $\Pi.\mathsf{Detect}(\kappa, \overline{x}, y)$. If the result is `false`, it outputs `invalid`. If the result is `true`, it outputs `tampered`.

Note that the decoder never outputs `valid`, and thus Γ does not satisfy correctness (which is not required here).

Let us first prove the soundness property. By contradiction, assume Γ is not sound. Then, there exists some $\hat{\gamma} \in \Sigma^n$ such that

$$\mathbb{P}[\Gamma.\mathsf{Dec}(sk, \hat{\gamma}) \neq \texttt{invalid}] = \mathbb{P}[\Pi.\mathsf{Detect}(\kappa, \overline{x}, \hat{\gamma}) = \texttt{true}] \geq 1/\mathsf{poly}(\lambda),$$

where the probability is over the choice of the secret key. This violates the soundness of the watermarking scheme.

It remains to prove tamper detection. By contradiction, assume Γ does not satisfy $\hat{\mathcal{F}}$-tamper detection. Then, there exists some (possibly randomized) function $\hat{f} \in \hat{\mathcal{F}}$ such that

$$\mathbb{P}[\Gamma.\mathsf{Dec}(sk, \tilde{\gamma}) \neq \texttt{tampered} \wedge \tilde{\gamma} \neq \gamma] = \mathbb{P}[\Pi.\mathsf{Detect}(\kappa, \overline{x}, \tilde{\gamma}) = \texttt{false} \wedge \tilde{\gamma} \neq \gamma] \geq 1/\mathsf{poly}(\lambda),$$

where $\tilde{\gamma} \leftarrow\!\!\$\ \hat{f}(\gamma)$ and where the probability is over the choice of the secret key, the randomness used to generate $\gamma \leftarrow\!\!\$\ \mathsf{Model}_\kappa(\overline{x})$, and the randomness of $\hat{f}$. This violates robustness of the watermarking scheme. □

When the output of the generative model is a fixed-length binary string (i.e., $\Sigma = \{0,1\}$ and $\mathcal{Y} = \{0,1\}^n$), the connection above becomes especially powerful. By mapping any robust watermarking scheme to a zero-bit tamper-detection code, we can directly apply our main impossibility result for codes to watermarking itself.

Specifically, our earlier theorem (Theorem 1) shows that no zero-bit tamper-detection code can be sound and robust to tampering beyond a certain threshold. By combining this with the reduction above, we obtain an immediate impossibility for robust watermarking of binary outputs:

Corollary 2. *Let $\mathcal{M} = \{\mathsf{Model} : \mathcal{X} \to \{0,1\}^n\}$ be any class of generative models with fixed-length binary outputs. There is no watermarking scheme for $\mathcal{M}$ that achieves both soundness and $\hat{\mathcal{F}}_\alpha^{\mathrm{ind}}$-robustness for any tampering rate $\alpha \geq (1+\delta)/2$, for any $\delta \in (0,1)$. This holds even if the model's prompt space $\mathcal{X}$ is arbitrary.*[3]

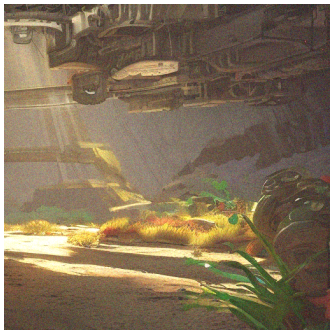
The same discussion also suggests an analogous limitation under a constant-bounded watermark soundness notion for any fixed $c \in (0,1)$, in the same informal sense discussed after Theorem 1. In particular, this rules out even a "useless" watermarking scheme whose soundness is achieved only with a high constant probability, such as $c = 1/2$.

The impossibility threshold we have identified for tamper-resilient codes is not just a technical detail; it serves as a universal limit for watermarking with fixed-length binary outputs. Any such watermarking scheme must face this same upper bound, no matter how the scheme is constructed or how prompts are chosen. The recent work of Christ and Gunn [CG24] provides a compelling example: their PRC-based watermarking schemes achieve robustness against independent bit-flip tampering up to the threshold $\alpha < (1-\delta)/2$, and their approach even accommodates certain types of deletions. Our corollary shows that this rate is not only achievable but also optimal. Although we do not claim a perfect correspondence between all watermarking schemes and all PRC constructions, since this depends on the precise model and tampering family, the upper bound on robustness holds for all such schemes.

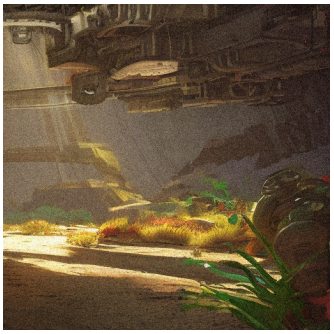
[3]For completeness, we highlight that it is possible to extend Corollary 2 to watermarking schemes supporting models with arbitrary output length (i.e., $\mathcal{M} = \{\mathsf{Model} : \mathcal{X} \to \{0,1\}^*\}$). This is because Theorem 1 holds even for tamper-detection codes with variable codeword length. We refer the reader to "Extension to weak soundness and variable length" paragraph in Section 3.2.



(a) Original Image



(b) Color Attack: 23% Error



(c) HSV Attack: 26% Error



(d) JPEG Attack: 32% Error



(e) WebP Attack: 34% Error

Figure 1: A series of images showing various attacks on a watermarked image, none of which removes the watermark

5 A Concrete Attack

5.1 Setup

Our goal is to test whether the theoretical robustness threshold we proved manifests in practice. For this purpose, we analyze the PRC watermark of Gunn, Zhao, and Song [GZS25], implemented in Stable Diffusion 2.1 Base (512×512 resolution, 50 denoising steps). This watermark is representative of the state of the art: it preserves perceptual quality, is provably undetectable under standard assumptions, and resists a wide class of natural manipulations.

The scheme embeds a PRC (pseudorandom) codeword in the latent space. Let $\mathbf{v} \in \mathbb{R}^{4\times64\times64}$ denote the latent tensor produced by the diffusion model. For each entry v_i of $\mathbf{v}$, the magnitude $|v_i|$ is preserved and the sign is set according to the corresponding codeword bit $\gamma_i \in \{0, 1\}$, where $\gamma = \gamma_1||\ldots||\gamma_n$ is the bit decomposition of a PRC codeword. As a result, the watermarked latent still follows the Gaussian distribution required for high-quality image generation. The codeword is pseudorandom, so without the detection key the modified latent is indistinguishable from a fresh Gaussian sample.

Detection proceeds by first inverting a generated image $\hat{x}$ back into a latent $\hat{\mathbf{v}}$ using an approximate inversion procedure, and then extracting sign information from $\hat{\mathbf{v}}$. Because inversion is lossy, $\hat{\mathbf{v}}$ typically agrees with the original latent on about 90% of entries even in the absence

of any edits. The scheme includes a detector and a separate belief-propagation (BP) decoding routine that attempts to recover the original codeword. The key parameter in our experiments is the pre-decoding sign error rate, namely the fraction of latent coordinates whose sign differs between the original watermarked latent $\mathbf{v}$ and the latent $\hat{\mathbf{v}}$ recovered from the attacked image before decoding. The latent dimension is $n = 4 \cdot 64 \cdot 64 = 16384$. As reported in [GZS25], the BP decoder is probabilistic and exhibits variable robustness: it sometimes recovers the codeword under substantial perturbations, but overall it is less reliable than the detector, which remains robust across a wide range of attacks. In our experiments we observe that once the pre-decoding error rate approaches one-half of the latent positions, the BP decoder fails consistently.

This threshold serves as the reference point for our experiments. The central question is whether simple image manipulations can drive the recovered latent to the 50% error mark, thereby erasing the watermark.

5.2 First Attempts

Before developing our attack, we tested the robustness of [GZS25] under a broad set of common image manipulations chosen to preserve image quality (including gaussian noise, gaussian blur, pixel-wise color shifts, and lossy format conversions).

Image manipulations. Gaussian noise, Gaussian blur, HSV perturbations, and attribute adjustments (saturation, hue, exposure, contrast) were applied to watermarked images. Even when these visibly degraded the images, the pre-decoding error rate never exceeded 26%, and BP reduced the error to below 1%. The watermark remained detectable in every case.

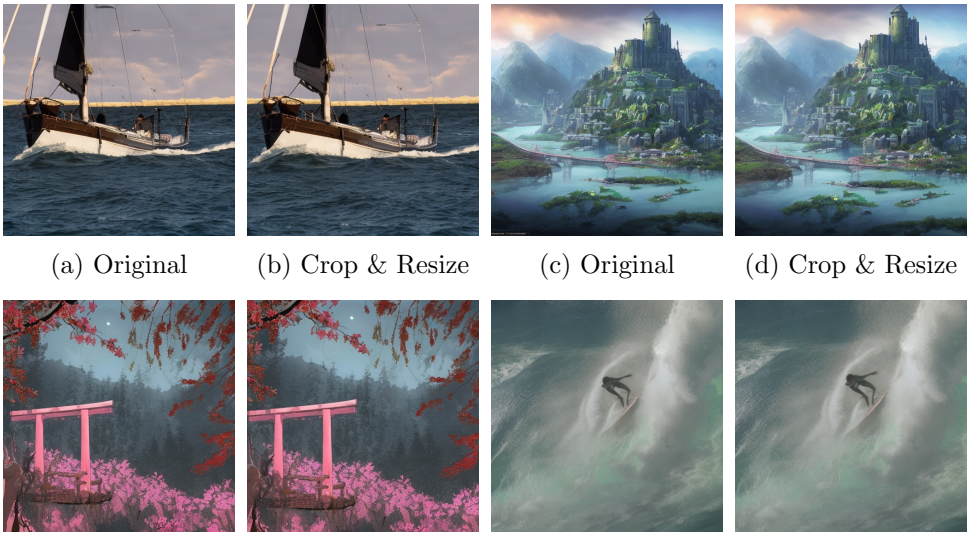
Pixel-wise color shifts. Next, we modified each pixel by adding fixed or randomized RGB offsets. A uniform shift of $(10, 0, 0)$, making the image slightly redder, produced under 10% error. A large uniform shift of $(75, 75, 75)$, which washed out the image, raised the error to only 12%. Randomized RGB perturbations across all pixels, with values drawn uniformly between 0 and 50, produced slightly higher disruption but never more than 23%. These results suggest that uniform color manipulations do not significantly disturb the sign pattern in latent space.

Lossy format conversions. Finally, we tested whether re-encoding the image in lossy formats could erase the watermark. JPEG compression at 15% quality produced heavy visible degradation and a 32% pre-decoding error rate, which BP reduced to 10%. WebP conversion at similar quality gave 34% error, reduced to 15% after decoding. In both cases, the watermark remained detectable.

5.3 Our Successful Attack

The failure of local manipulations suggests what is missing. Noise, blur, color shifts, and compression alter pixel values but preserve the coordinate system of the image. The encoder still sees essentially the same structure, and so the latent sign pattern (i.e., the PRC codeword γ) drifts only modestly. To break the watermark, we sought an operation that would force the encoder to reinterpret the image globally, while leaving the picture perceptually unchanged.

This reasoning led to the crop-and-resize attack. We take a watermarked 512×512 image, crop 15 pixels from each side to obtain a 482×482 image, and then resize it back to 512×512 using bicubic interpolation. The crop removes only a narrow border, and the interpolation smoothly reconstructs missing pixels from their neighbors. Visually, the result is nearly indistinguishable from the original. We use a 15 px crop by default because it is more reliable



(a) Original (b) Crop & Resize (c) Original (d) Crop & Resize

(e) Original (f) Crop & Resize (g) Original (h) Crop & Resize

Figure 2: Comparison between original and crop & resize attacked images.

across examples, although for tightly framed images a 10 px crop still suffices, as in Figure 6. A 5 px crop is no longer reliable. Bilinear interpolation shows the same qualitative behavior as bicubic interpolation. Examples are shown in Figure 2, where side-by-side comparisons reveal no perceptible difference between the original and the cropped-and-resized images.

The effect on the watermark is decisive. Across more than one thousand test cases, the recovered latent after crop-and-resize showed pre-decoding error rates concentrated near 50%. This error rate is exactly the threshold at which BP decoding fails. Indeed, in every trial the decoder was unable to recover the codeword, and detection failed completely. Unlike the earlier attacks, which either left the watermark intact or visibly degraded the image, this single transformation erased the watermark while keeping the content unchanged.

In short, crop-and-resize is the only edit we found that preserves perceptual quality while driving the latent representation to the theoretical boundary. The watermark disappears, not because the image is damaged, but because the latent has been resampled in a way that overwhelms the error-correcting capacity of the code.

5.4 Why It Works

The success of crop-and-resize can be traced to how the encoder interprets the image. Local edits (noise, blur, color shifts, compression) perturb pixel values but leave the grid of the image intact. The encoder still recognizes the same arrangement, and the latent sign pattern changes only in scattered places. Cropping followed by resizing is different. Cropping removes a thin border, and resizing does not simply stretch the remaining pixels. It recomputes every pixel value by bicubic interpolation on a new lattice. The encoder is then asked to describe the same visual content in a new coordinate system. This global resampling is what drives nearly half of the latent entries across zero, flipping their signs.

Latent-space effects. Figure 3 shows this effect in detail. The original latent sign pattern (top left) encodes the watermark (i.e., the PRC codeword) as a pseudorandom arrangement of signs. After crop-and-resize, the recovered latent (bottom left) is visibly scrambled. The difference map before decoding (top right) shows that 48.07% of the signs have flipped. This

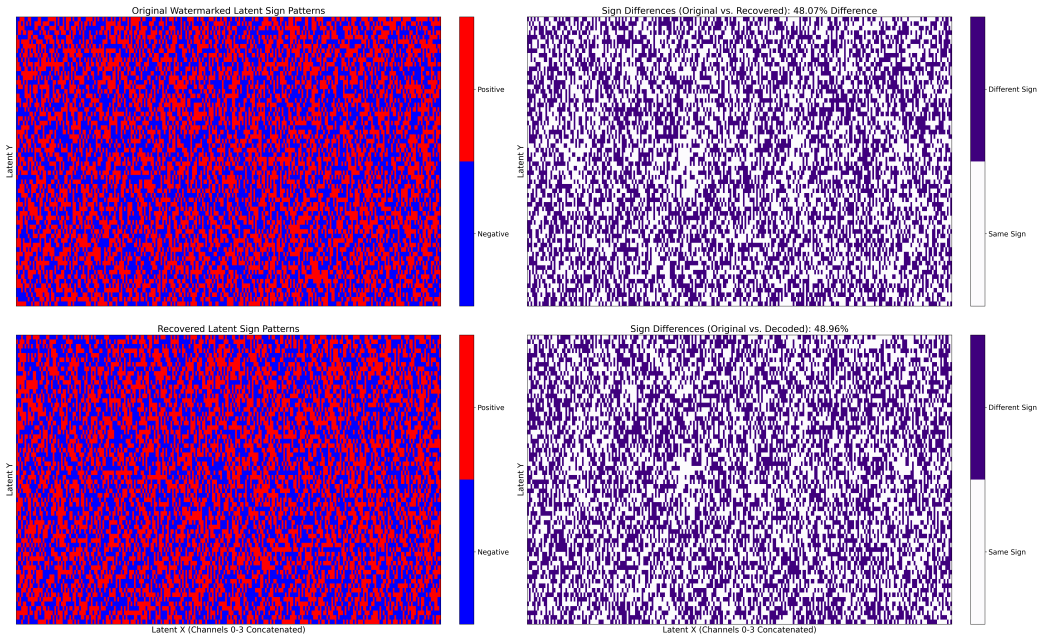


Figure 3: Latent sign analysis of a watermarked image before and after the crop and resize attack. Top left: the original latent sign pattern (red = positive, blue = negative) that encodes the watermark as a pseudorandom codeword. Bottom left: the latent recovered from the attacked image, which is visibly scrambled compared to the original. Top right: a difference map comparing the original and recovered latents, showing that 48.07% of signs have flipped (purple = flipped, white = unchanged). Bottom right: a difference map after belief propagation, the error-correcting decoder, has tried to repair the errors. Because the error rate is already near the 50% threshold, the decoder fails and converges to a different pseudorandom codeword. The resulting error rate increases slightly to 48.96%, confirming that the watermark cannot be recovered.

is essentially the information-theoretic boundary: once errors approach 50%, the codeword is indistinguishable from random. Of course, crop-and-resize is a structured image transformation rather than the i.i.d. corruption channel of Theorem 1. The point is that, empirically, it pushes the recovered sign pattern close to the same critical threshold at which the codeword becomes effectively indistinguishable from random. The bottom-right panel shows the result after belief propagation, the error-correcting decoder, has attempted to repair the errors. At this level of corruption, BP cannot recover the original codeword and instead converges to a different pseudorandom codeword. As a result, the measured error rises slightly to 48.96%. This increase is not paradoxical but diagnostic: the decoder has lost all correlation with the true codeword. The image itself is unchanged, but the watermark is irretrievable.

Variants. Other global manipulations show the same pattern. Figure 4 compares crop and resize with several related operations. Cropping and padding with black pixels produced about 16.7% error and left a conspicuous border. Downscaling and then upscaling, even aggressively to 312×312 before resizing back, produced about 12.1% error. Neither came close to the threshold. Downscaling and padding with black pixels did reach $\sim 50\%$ error (Figure 5), but the added border made the alteration visually obvious, as shown in subfigure (d) of Figure 4.



(a) Original Image



(b) Crop & Resize (47.2% error)



(c) Crop & Pad (16.7% error)

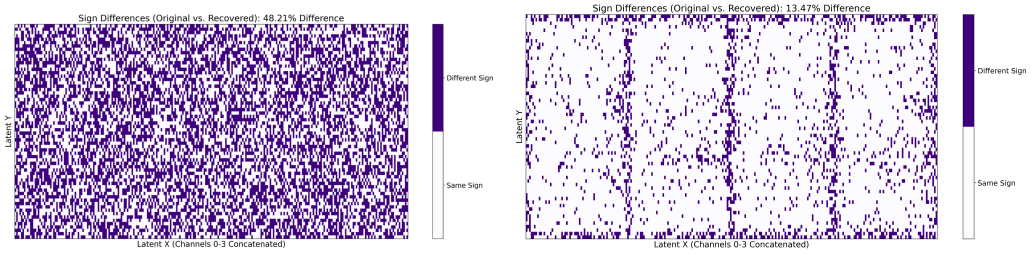


(d) Downscale & Pad (49.8% error)



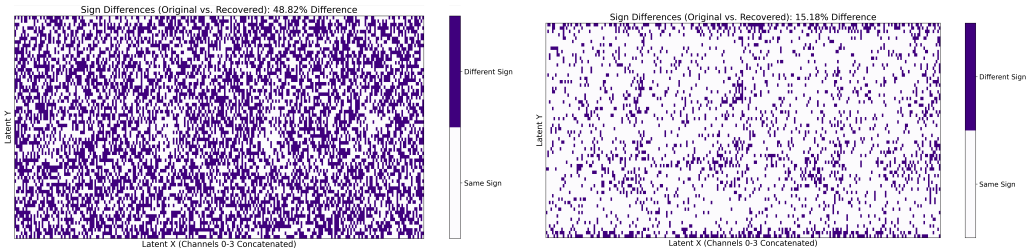
(e) Downscale & Upscale (12.1% error)

Figure 4: Visualization of this branch of attacks (b-e) compared to the original (a)



(a) Crop & Resize

(b) Crop & Pad



(c) Downscale & Pad

(d) Downscale & Upscale

Figure 5: Side-by-side comparison of latent space visualizations for the main attack and each variant

In contrast, crop and resize combined two properties: it perturbed the latent enough to destroy the watermark, and it did so while preserving the appearance of the image.



(a) Original (b) Cropped (10px)

Figure 6: Watermark removal on bordered image.

Generalization. The vulnerability is not limited to this specific setup. Generative models often allow users to control framing explicitly. An adversary can request an image with an artificial border and then remove it by cropping and resizing. Figure 6 illustrates this approach: the model produces an image with a frame, the border is removed, and the watermark disappears while the content remains intact. No knowledge of the watermark key is needed, and no optimization is required. The broader lesson is that transformations that globally resample the image while keeping their perceptual content largely unchanged can push the latent toward the robustness threshold.

Data and Code Availability

Our theoretical results are self-contained. The empirical study in Section 5 uses only synthetic images generated by Stable Diffusion together with the publicly described PRC-based watermark of Gunn, Zhao, and Song [GZS25]. No proprietary data or data from human subjects is used.

To support reproducibility, we provide a repository at `https://github.com/ygoonati/Robust-Watermarking-Limits`. The repository contains:

- the code that interfaces with the watermark implementation and applies the crop-and-resize transformation;
- configuration files for the Stable Diffusion pipeline and prompts used in our experiments;
- scripts that reproduce the figures and numerical results in the paper from a fresh checkout.

These artifacts are sufficient to re-run the experiments and verify the empirical claims in the paper.

Acknowledgments

Danilo Francati was supported by project SERICS (PE00000014), under the MUR National Recovery and Resilience Plan funded by the European Union- NextGenerationEU. Daniele Venturi

is member of the Gruppo Nazionale Calcolo Scientifico Istituto Nazionale di Alta Matematica (GNCS-INdAM). His research was partially supported by project SERICS (PE00000014) and by project PARTHENON (B53D23013000006)—under the MUR National Recovery and Resilience Plan funded by the European Union-NextGenerationEU—by project BEAT—funded by Sapienza University of Rome—and by the AI-Sec Lab initiative between Sapienza University of Rome and the CYBER 4.0 Competence Center.

References

[AAC+25] Omar Alrabiah, Prabhanjan Ananth, Miranda Christ, Yevgeniy Dodis, and Sam Gunn. Ideal pseudorandom codes. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 1638–1647, New York, NY, USA, 2025. Association for Computing Machinery.

[Aar22] Scott Aaronson. My AI safety lecture for UT effective altruism. https://scottaaronson.blog/?p=6823, 2022. Discusses watermarking projects at OpenAI. Accessed: September 2025.

[ADR+24] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. Waves: benchmarking the robustness of image watermarks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[ARC+01] Mikhail J. Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In Ira S. Moskowitz, editor, *Information Hiding*, pages 185–200, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[ARH+03] Mikhail J. Atallah, Victor Raskin, Christian F. Hempelmann, Mercan Karahan, Radu Sion, Umut Topkara, and Katrina E. Triezenberg. Natural language watermarking and tamperproofing. In Fabien A. P. Petitcolas, editor, *Information Hiding*, pages 196–212, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[Arn00] Michael Arnold. Audio watermarking: Features, applications and algorithms. In *2000 IEEE International conference on multimedia and expo. ICME2000. Proceedings. Latest advances in the fast changing world of multimedia (cat. no. 00TH8532)*, volume 2, pages 1013–1016. IEEE, 2000.

[BOD95] F.M. Boland, J.J.K. O'Ruanaidh, and C. Dautzenberg. Watermarking digital images for copyright protection. In *Fifth International Conference on Image Processing and its Applications, 1995.*, pages 326–330, 1995.

[BTH96] Laurence Boney, Ahmed H Tewfik, and Khaled N Hamdy. Digital watermarks for audio signals. In *Proceedings of the third IEEE international conference on multimedia computing and systems*, pages 473–480. IEEE, 1996.

[CG24] Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VI*, volume 14925 of *LNCS*, pages 325–347. Springer, Cham, August 2024.

[CGLS25] Yixin Cheng, Hongcheng Guo, Yangming Li, and Leonid Sigal. Revealing weaknesses in text watermarking through self-information rewrite attacks. In *Forty-second International Conference on Machine Learning*, 2025.

[CGZ24] 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.

[CMB+07] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., 2007.

[Eur24] European Parliament and Council of the European Union. Regulation (EU) 2024/1689: Artificial Intelligence Act. https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng, 2024. Official Journal of the European Union, L 1689, 12 July 2024. Accessed: September 2025.

[Exe23] Executive Office of the President of the United States. Executive Order No. 14110: Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence. https://www.federalregister.gov/documents/2023/11/01/2023-24283, 2023. Federal Register, Vol. 88, No. 212, pp. 75191–75212. Accessed: September 2025.

[FCJ+23] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023.

[FGJ+25] Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. Publicly-detectable watermarking for language models. *IACR Communications in Cryptology*, 1(4), 2025.

[FXTB22] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. Supervised gan watermarking for intellectual property protection. In *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2022.

[GG24] Surendra Ghentiyala and Venkatesan Guruswami. New constructions of pseudorandom codes. *arXiv preprint arXiv:2409.07580*, 2024.

[GZS25] Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[HD17] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1951–1960, Red Hook, NY, USA, 2017. Curran Associates Inc.

[KGW+23] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023.

[KH25] Andre Kassis and Urs Hengartner. Unmarker: A universal attack on defensive image watermarking. In *2025 IEEE Symposium on Security and Privacy (SP)*, 2025.

[KSK+23] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500, 2023.

[KTHL24] Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024.

[LDFK24] Nils Lukas, Abdulrahman Diaa, Lucas Fenaux, and Florian Kerschbaum. Leveraging optimization for adaptive attacks on image watermarks. In *The Twelfth International Conference on Learning Representations*, 2024.

[RP97] Joseph Ó Ruanaidh and Thierry Pun. Rotation, scale and translation invariant digital image watermarking. *Proceedings of International Conference on Image Processing*, 1:536–539 vol.1, 1997.

[SSR+24] Mehrdad Saberi, Vinu Sankar Sadasivan, Keivan Rezaei, Aounon Kumar, Atoosa Chegini, Wenxiao Wang, and Soheil Feizi. Robustness of AI-image detectors: Fundamental limits and practical attacks. In *The Twelfth International Conference on Learning Representations*, 2024.

[WKGG23] 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.

[WWZ+22] Wenbo Wan, Jun Wang, Yunming Zhang, Jing Li, Hui Yu, and Jiande Sun. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488:226–247, 2022.

[YSAF21] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14428–14437, 2021.

[ZEF+23] Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. *arXiv preprint arXiv:2311.04378*, 2023.

[ZEF+24] Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: impossibility of strong watermarking for language models. In *Forty-first International Conference on Machine Learning*, 2024.

[ZKJFF18] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, page 682–697, Berlin, Heidelberg, 2018. Springer-Verlag.

[ZPD+23] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

[ZZS+24] 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. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 8643–8672. Curran Associates, Inc., 2024.

[ZZXP23] Yu Zeng, Mo Zhou, Yuan Xue, and Vishal M Patel. Securing deep generative models with universal adversarial signature. *arXiv preprint arXiv:2305.16310*, 2023.

A A Construction of Zero-bit Tamper-detection Code

In this appendix we give an explicit information-theoretic construction of a zero-bit tamper-detection code. Our purpose is not to propose a practical scheme, but to show that the robustness threshold from Section 3.2 is tight: there are codes that achieve soundness and tamper detection for any corruption rate strictly below the critical value. The construction is deliberately simple. The secret key itself serves as the codeword, and the decoder classifies a candidate string according to its Hamming distance from the key. This partitions the space of strings into three regions corresponding to `valid`, `tampered`, and `invalid`, separated by a distance threshold t.

We now formalize this construction.

Construction 1. Let n be the codeword length, and fix a parameter $\delta \in (0,1)$. Define $t = n(1-\frac{1}{q})(1-\delta)$. The code $\Gamma = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as:

Key Generation $\mathsf{KGen}(1^\lambda)$: Sample a secret key $sk \leftarrow\!\!\$\ \Sigma^n$ uniformly at random, where $|\Sigma| = q \geq 2$.

Encoding $\mathsf{Enc}(sk)$: Output the codeword $\gamma = sk$.

Decoding $\mathsf{Dec}(sk, \gamma)$: Given $\gamma \in \Sigma^n$:

- If $\mathrm{dist}(\gamma, sk) > t$, output `invalid`.
- If $0 < \mathrm{dist}(\gamma, sk) \leq t$, output `tampered`.
- If $\gamma = sk$, output `valid`.

The correctness property for this construction is immediate: the decoder always accepts the honest codeword, which is just the secret key.

What remains are the two core security properties: soundness, and robust detection of tampering up to the optimal threshold. Both follow from a simple analysis of Hamming distance and concentration.

Theorem 3 (Security of the Simple Construction)**.** *Let $\delta \in (0,1)$ be a constant, and let n and $q = |\Sigma|$ such that $\frac{n}{q} \in \omega(\log \lambda)$.* [4] *The zero-bit tamper-detection code Γ from Construction 1 satisfies soundness and $\mathcal{F}_{\alpha n}$-tamper detection for $\alpha = (1-\frac{1}{q})(1-\delta)$.*

[4] This can always be enforced by increasing the codeword length n with respect to the cardinality q of Σ.

Proof. We prove each property in turn.

Soundness: Fix any string $\hat{\gamma} \in \Sigma^n$ and random key $sk \leftarrow\!\!\$\ \Sigma^n$. Let X be the number of positions where $\hat{\gamma}[i] = sk[i]$; X is binomial with mean n/q. The decoder outputs `invalid` unless

$$X > n - t = n - n\Big(1 - \frac{1}{q}\Big)(1 - \delta) \geq \frac{n}{q}(1 + \delta),$$

where the last inequality holds whenever $q \geq 2$.

By the Chernoff bound,

$$\mathbb{P}\Big[X \geq \frac{n}{q}(1 + \delta)\Big] \leq \exp\Big(-\frac{\delta^2 n}{3q}\Big),$$

which is negligible whenever $\frac{n}{q} \in \omega(\log \lambda)$ and $\delta > 0$ is constant.

Tamper Detection: Fix any $f \in \mathcal{F}_{\alpha n}$ with $\alpha = (1 - \frac{1}{q})(1 - \delta)$. Let $\gamma = sk$ and $\tilde{\gamma} = f(\gamma)$, with $\tilde{\gamma} \neq \gamma$. By definition of f, $\mathrm{dist}(\tilde{\gamma}, sk) \leq n(1 - \frac{1}{q})(1 - \delta) = t$. Thus, the decoder outputs `tampered`.

Both properties hold as claimed. □