SELFCHECKMARK: CONFIDENCE-DRIVEN DUAL-CHANNEL WATERMARKING AND SELF-CORRECTING EXTRACTION FOR VIDEO DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

As video diffusion models gain wider use and generated videos become more prevalent, reliable copyright attribution and provenance tracing become increasingly important. In-generation watermarking addresses this need by embedding information during synthesis. A prominent route modifies the initial diffusion noise and recovers the embedded watermark information through video inversion. Yet generation, video distortions, and inversion errors yield recovered latent evidence of uneven reliability, so weak observations can dilute strong ones. Moreover, correcting an inverted noise estimate before combining redundant observations can fail when its errors exceed the code's tolerance. To address these challenges, we propose SelfCheckMark, a blind video watermarking framework with confidence-aware extraction and post-fusion self-correction. During embedding, extended Hamming(8,4) coding and a dual-channel design assign each coded bit to a strong-response anchor and continuous redundant elements. During extraction, local values are summed so elements with larger absolute values contribute more, while channel–frame replicas are weighted by inverse standard deviation, reducing the influence of unreliable evidence. Only after replica fusion is the most likely legal Hamming codeword selected, allowing correction to act on the combined evidence. Experiments show that, with a 512-bit watermark, SelfCheckMark achieves 100% mean bit accuracy on clean videos, over 98% under Gaussian noise and blur, and 85.36% under H.264 compression. Ablations confirm both confidence fusion and post-fusion correction.

1 INTRODUCTION

Video diffusion models have lowered the barrier to high-quality video generation. Large-scale models such as HunyuanVideo can produce long videos with high visual quality, motion coherence, and semantic consistency (Kong et al., 2024; Wu et al., 2025; Lin et al., 2026). At the same time, the ease of producing and distributing generated videos raises questions of copyright attribution, content provenance, and platform governance. These concerns call for reliable ways to identify the origin of generated content. Digital watermarking offers one approach by embedding recoverable information into generated content.

Existing watermarking methods for video diffusion models fall into two categories: post-processing methods and in-generation watermarking methods. Post-processing methods embed watermark information into a rendered image or video, for example through learned encoders or signal-domain transforms (Zhu et al., 2018; Tancik et al., 2020; Luo et al., 2023). They can be used with different generators, but must balance watermark recovery against perceptual changes to the output. In-generation watermarking methods instead incorporate the watermark into synthesis, either by modifying a model component or by shaping the initial noise of a diffusion model (Fernandez et al., 2023; Wen et al., 2023; Yang et al., 2024). Initial-noise watermarking is a mainstream in-generation approach for video diffusion models. Its extraction typically inverts the generated video to estimate the noise carrying the watermark information. Recent video methods also account for repeated observations, frame correspondence, or key-based recovery of watermark information (Hu et al., 2025a;b; Zhao et al., 2026). Nevertheless, because the original noise is only approximately recovered, the way these methods use inverted latent evidence remains central to reliable extraction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Generation, video distortions, and inversion introduce errors of varying strength into this recovered noise, creating two unresolved problems: **(1) Unequal reliability of inverted observations.** Some latent values strongly support a bit while others are ambiguous or misleading. These differences occur within local regions and across channel–frame replicas, so equal voting lets unreliable observations dilute stronger evidence. **(2) Correction before redundant evidence is combined.** Error-correcting codes have finite tolerance when applied directly to an inverted noise estimate. If inversion errors exceed that tolerance, correcting the estimate before combining repeated observations cannot reliably recover the watermark information. Fusing the available evidence first can reduce the error burden presented to the code.

To address these limitations, we propose SelfCheckMark, a blind watermarking framework illustrated in Figure 1. During watermark generation and embedding, we encode the watermark information into extended Hamming codewords, distribute their bits across strong-response anchors and continuous redundant elements, and scramble signs with a global key. This supplies repeated bit observations while preserving random sign statistics. During extraction, inversion provides a noise estimate; values within each local region are combined without binarization, and channel–frame replicas are weighted by inverse standard deviation. This two-level fusion reduces the influence of uncertain observations. After fusion, maximum-likelihood self-correction selects a legal Hamming codeword from the combined evidence, allowing redundant observations to inform the final decision. Model parameters remain fixed, and extraction requires neither the original video nor a per-video template.

The main contributions of this paper are as follows:

1. **Confidence-aware watermarking.** We propose SelfCheckMark, a confidence-driven dual-channel watermarking framework for video diffusion models. It builds sparse anchors from strong-response elements and a continuous redundancy channel from the remaining elements, and fuses latent magnitudes with inverse-standard-deviation replica weighting at extraction, avoiding equal weighting of unreliable evidence in the inverted noise.
2. **Post-fusion self-correction.** We apply the extended Hamming(8,4) codeword constraint only after replica fusion, so redundant evidence informs the final correction step. It recovers the 512-bit watermark information by maximum-likelihood selection over the 16 legal codewords of each block, using polarity and magnitude to reduce residual bit errors under video distortions.
3. **Robustness and ablations.** Extensive experiments demonstrate that SelfCheckMark attains superior or competitive watermark recovery under Gaussian noise, H.264 compression, and Gaussian blur, while ablations verify the contribution of each core module.

2 Related Work

2.1 Post-Processing and In-Generation Watermarking

Early deep watermarking methods, building on classical spread-spectrum designs (Cox et al., 1997), embed signals in the pixel or transform domain and learn robustness through end-to-end training under simulated attacks: HiDDeN (Zhu et al., 2018) couples an information encoder with a differentiable noise layer, StegaStamp (Tancik et al., 2020) targets physical print-and-capture, and DVMark (Luo et al., 2023) exploits multi-scale spatio-temporal structure for video. These post-processing methods are model-agnostic but require an additional encoder and trade off pixel distortion with information recovery. With diffusion models, in-generation watermarking embeds the signal within the generation process: Stable Signature (Fernandez et al., 2023) fine-tunes the decoder of a latent diffusion model (Rombach et al., 2022), Tree-Ring (Wen et al., 2023) writes patterns into the frequency domain of the initial noise, and Gaussian Shading (Yang et al., 2024) keeps the watermarked latents statistically consistent with the standard Gaussian; ROBIN (Huang et al., 2024a) instead optimizes the watermark adversarially for robustness and invisibility. Post-processing can perturb the rendered output, while inversion-based initial-noise methods rely on recovered latents whose reliability varies after video generation and distortion. SelfCheckMark embeds in the initial noise with fixed model parameters and fuses continuous recovered evidence locally and across channel–frame replicas to reduce the influence of unreliable observations.

2.2 Video Diffusion Watermarking and Error Correction

For video, diffusion models (Ho et al., 2022) carry this paradigm into the temporal dimension. VideoShield (Hu et al., 2025a) maps template bits onto the initial noise of video diffusion and extracts them through DDIM inversion; VideoMark (Hu et al., 2025b) uses frame-level pseudo-random error-correcting codes and a temporal matching module; SIGMark (Zhao et al., 2026) binds the watermark to a global frame-level key to achieve blind extraction without per-video auxiliary information; and SPDMark (Fares et al., 2026) embeds the watermark through selective parameter displacement. These methods address template construction, temporal alignment, key management, and parameter-level embedding. Their extraction procedures also differ in the evidence retained after inversion. VideoShield compares recovered noise with a reference template, while SIGMark uses a global key without per-video template storage. Error-correcting codes add structured redundancy to information bits, and soft-decision rules can also use observation reliability (Hamming, 1950; Chase, 1972; Forney, 1966). Per-bit hard decisions, however, discard latent magnitudes before correction. SIGMark applies PRC-based correction (Christ & Gunn, 2024) to the signs of each inverted latent frame, so a large frame-level error can exceed the code's tolerance before other observations are combined. These procedures can therefore discard reliability information or require correction on a frame whose errors exceed the code's tolerance. SelfCheckMark retains local magnitudes, weights channel–frame replicas by reliability, and applies maximum-likelihood Hamming self-correction only after fusion.

3 Method

Figure 1 shows the full SelfCheckMark pipeline. During watermark generation, the watermark information is encoded into legal Hamming codewords, mapped to the initial noise by dual-channel embedding, and sign-scrambled with a global key. The diffusion model then produces a watermarked video without parameter changes or post-processing. During extraction, the query video is inverted to estimate the initial noise. After keyed sign descrambling, continuous observations are fused within local blocks and across channel–frame replicas; only then is the watermark information recovered by a codeword-level decision. The global key enables blind extraction, while this ordering allows redundant observations to inform the final correction.

3.1 Problem Definition

Let $m \in \{0,1\}^{512}$ be the watermark information, Z the initial noise, and K the global key; the information fields are specified in Appendix A. The `inf` labels in Figure 1 refer to this watermark information, including m and its encoded representations. Embedding maps the encoded watermark information into Z and uses K to produce watermarked initial noise Z_w. Given image condition I and text condition T, the diffusion model $\mathcal{G}_\theta$ generates $V_w = \mathcal{G}_\theta(I, T, Z_w)$. A distortion operator A represents processing of the generated video and produces the query $V_q = A(V_w)$. At extraction, the public inversion model estimates the watermarked noise from V_q; the keyed mask is reproduced, the observations are fused, and the extractor outputs $\hat{m} = D(V_q; K)$. It uses neither the original video nor a per-video template or key, so extraction remains blind. Because inversion need not reproduce Z_w exactly, D treats recovered signs and magnitudes as uncertain observations rather than fixed bit decisions. The central task is to combine those observations before applying the codeword constraint.

3.2 Self-Correcting Watermark Information Generation

Errors in the recovered noise can make individual watermark bits uncertain. We therefore encode the watermark information into legal codewords before embedding, providing redundancy that the extractor can use after local and replica evidence have been fused. Encoding supplies the codeword constraint; correction is deferred to Section 3.5 rather than applied to each inverted estimate. The minimum-distance guarantee applies to a hard codeword with at most one erroneous bit; it does not make a severely corrupted inverted frame correctable. Replica fusion instead supplies aggregated evidence for the final codeword decision.

We divide the 512-bit watermark information into 128 four-bit blocks $u_j \in \{0,1\}^4$ and encode each into an 8-bit codeword $C_j = u_j G \pmod 2$ with the extended Hamming(8,4) code, whose minimum

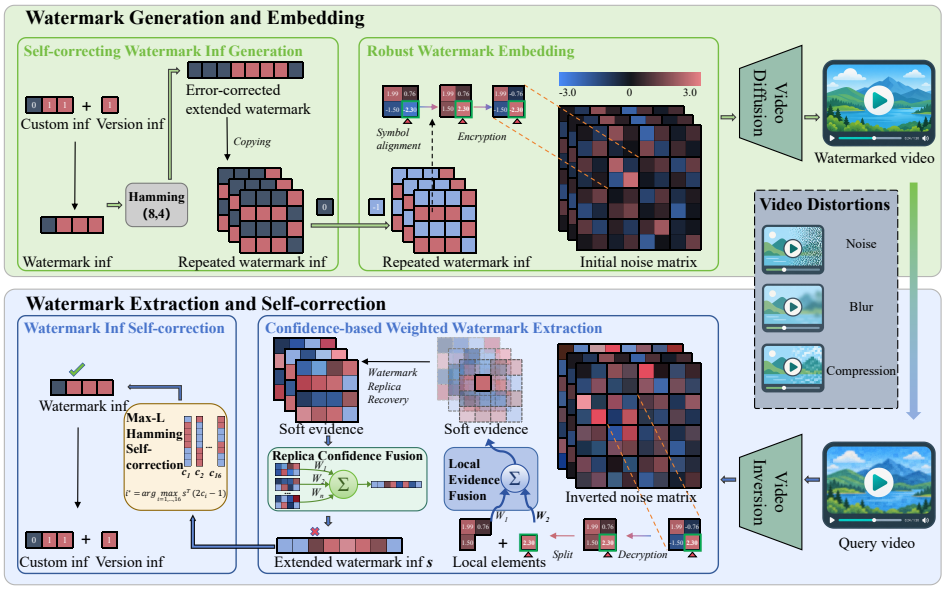

Figure 1: Overview of SelfCheckMark, including watermark generation and embedding, confidence-weighted extraction, and self-correction. Here, "inf" stands for "information."

distance 4 corrects any single-bit error; the generator matrix G and the matching parity-check matrix H are given in Appendix A. Concatenating the 128 codewords yields a 1024-bit extended stream $c = \text{concat}(C_1, \ldots, C_{128}) = [c_1, \ldots, c_{1024}]$. Each bit is mapped to a bipolar value $b_i = 2c_i - 1 \in \{-1, +1\}$ to match the sign form of the latents, giving the target sign sequence $b = [b_1, \ldots, b_{1024}]$. This $2\times$ redundancy enables the codeword-level joint decision of Section 3.5.

3.3 Robust Watermark Embedding

Embedding must give each coded bit several observations without exposing a deterministic sign pattern in the initial noise. We use local blocks and channel–frame replicas to provide repeated evidence, pair a strong-response anchor with continuous redundant elements within each block, and then scramble element signs with the global key. The anchor contributes a strong local response, while the other elements retain continuous evidence that sign-only extraction would discard. The replicated layout gives the extractor multiple observations of each coded bit to weight after inversion. Concretely, the 1024 bipolar symbols are written into $Z \in \mathbb{R}^{1\times16\times16\times64\times64}$. The latent has 16 channels of 16 latent frames, each a 64×64 plane; we divide every plane into non-overlapping 2×2 blocks, giving $64^2/2^2 = 1024$ blocks one-to-one with the codeword bits, and the i-th block carries b_i. Each block is repeated across all $R = 16\times16 = 256$ channel–frame replicas, so every bit is carried by $4\times256 = 1024$ latent elements, giving a usable empirical distribution for confidence modeling at extraction. Within each block, the element of largest absolute magnitude is the strong-response "anchor" and the remaining three form the continuous redundancy channel; both carry the same bit and are fused at extraction. All four elements are then sign-aligned to the target bit, $z_{r,i,p} \leftarrow b_i \cdot |z_{r,i,p}|$, where $z_{r,i,p}$ denotes the element of Z at replica r, block i, and position p; this only flips signs and preserves the original magnitude distribution.

Directly writing this way would make each block's four elements share one sign, a locally detectable structure. We therefore apply keyed per-element sign scrambling: using the global key K and a fixed nonce, a per-element binary mask $M \in \{+1, -1\}$ is generated from the ChaCha20 keystream (Nir & Langley, 2018) and applied element-wise as $z_{r,i,p} \leftarrow M_{r,i,p} \cdot z_{r,i,p}$, giving the watermarked initial noise Z_w. The mask is approximately uniform and independent, so the marginal signs become random again and the block structure is suppressed; extraction reproduces M from the same key and nonce. One clean first frame is kept for image conditioning and discarded at extraction.

3.4 CONFIDENCE-BASED WEIGHTED WATERMARK EXTRACTION

The goal of the extraction side is to recover the 512-bit watermark information from a possibly attacked query video. Rather than treating the inverted latents as equally reliable discrete templates, we preserve their continuous magnitudes, estimate the reliability of evidence at both the local-element and the replica level, and pass the two-level weighted result to the codeword-level decision. The fused scores retain both polarity and confidence for self-correction.

3.4.1 VIDEO INVERSION AND DECRYPTION

Given a query video V_q, we map it to the latent space with the VAE encoder of HunyuanVideo-I2V and estimate the initial noise by a 50-step FlowMatch Euler reverse process with an empty text prompt, $\hat{Z} = \mathrm{Inv}(\mathrm{Enc}(V_q); \varnothing)$. After removing the watermark-free first frame, we obtain $\hat{Z}_w \in \mathbb{R}^{1\times16\times16\times64\times64}$, one-to-one with the watermarked latent frames. Since V_q passed through the distortion operator A, $\hat{Z}_w$ deviates from Z_w element-wise: signs are partly flipped and magnitudes attenuated to different degrees, which is exactly what motivates the confidence model. We regenerate the mask M with the same key and nonce and descramble element-wise, $Z' = \hat{Z}_w \circ M$, restoring the sign direction aligned with the codeword bits while preserving continuous magnitudes. Hereafter $z'_{r,i,p}$ denotes the element of Z' at replica r, block i, position p.

3.4.2 LOCAL EVIDENCE FUSION

For the i-th codeword bit of the r-th replica, we sum the four descrambled soft values in the corresponding 2×2 block to obtain that replica's local evidence:

$$q_{r,i} = \sum_{p=1}^{4} z'_{r,i,p} \tag{1}$$

Since each element is a sign times an absolute magnitude, this summation accumulates magnitudes along each element's sign direction: large-magnitude elements contribute more to $q_{r,i}$, while elements close to zero contribute little. It therefore fuses the sparse anchor and the continuous redundancy channel into one scalar and performs a magnitude-weighted soft decision instead of a sign-only vote. Section 4.4.1 shows that quantizing block elements to signs gives them equal weight and loses magnitude information, with the largest observed penalty under H.264 compression.

3.4.3 REPLICA CONFIDENCE FUSION

The reliability of the 256 replicas is not identical: video content, compression strength, and inversion errors make the local evidence of some replicas more stable and that of others more variable, so the second level assigns them different weights.

We measure the instability of a replica by the dispersion of its internal local evidence, and choose the standard deviation as the concrete form of this measure. For the r-th replica, its local evidence sequence is $q_{r,1}, \ldots, q_{r,1024}$, with standard deviation:

$$\sigma_r = \mathrm{Std}(q_{r,1}, \ldots, q_{r,1024}) \tag{2}$$

A larger σ_r means more unstable observations, so this replica should receive a smaller weight. We accordingly construct normalized weights:

$$w_r = \frac{\omega_r}{\sum_{k=1}^{R} \omega_k}, \quad \omega_r = \frac{1}{\sigma_r + \varepsilon}, \quad R = 256 \tag{3}$$

where $\varepsilon = 1\times10^{-6}$ prevents numerical instability from zero variance. Normalization acts only on the 256 replica weights; both $q_{r,i}$ and s_i keep their original magnitude scale, so the decision can still exploit continuous confidence.

Using the above weights, the final fused soft evidence of the i-th codeword bit is:

$$s_i = \sum_{r=1}^{R} w_r \, q_{r,i}, \quad R = 256 \tag{4}$$

Concatenating the 1024 fused values in order yields the complete soft codeword $s = [s_1, \ldots, s_{1024}]$. Unlike the target sign sequence b, its elements are continuous: the sign gives the bit tendency and the absolute value the confidence strength. This continuous representation is the prerequisite for the codeword-level decision of Section 3.5 and is the key difference from per-bit hard decision.

3.5 Watermark Information Self-Correction

Deciding bit by bit from the sign of s would discard, at the last step, the confidence accumulated by the previous two levels of weighting, whereas the codeword structure of Section 3.2 provides usable constraints. The final stage therefore does not quantize before correcting; it performs joint inference on the continuous soft codeword directly in the space of legal codewords.

We divide s into 128 soft blocks of 8 bits; the j-th block $s^{(j)} = [s_{8(j-1)+1}, \ldots, s_{8j}]$ corresponds to the j-th extended codeword. The 4-bit input of the extended Hamming(8,4) code has only 16 values, so its legal codewords can be fully enumerated as the codebook $\mathcal{C} = \{c^{(1)}, \ldots, c^{(16)}\}$, whose values are given in Table 5 of Appendix A. For the j-th soft block we define its matching score with the a-th legal codeword as the inner product of the soft block and the corresponding bipolar codeword, which measures the magnitude-weighted sign agreement between the two. We then choose the legal codeword with the largest matching score:

$$a_j = \underset{a \in \{1,\ldots,16\}}{\arg\max} \left(s^{(j)}\right)^{\top} \left(2c^{(a)} - 1\right) \tag{5}$$

Through the information-to-codeword mapping stored in the codebook, a_j is looked up back to the 4-bit information block $\hat{u}_j$; deciding and concatenating the 128 soft blocks in order gives the recovered watermark information $\hat{m} = [\hat{u}_1, \ldots, \hat{u}_{128}] \in \{0,1\}^{512}$.

The essential difference from per-bit hard decision lies in the decision space: the latter binarizes each bit and corrects locally via the syndrome, and the magnitude lost at quantization cannot be recovered afterwards, whereas the above approach performs joint inference on 8 bits so that the preserved magnitudes keep participating in the decision. From a statistical viewpoint, if the bipolar codeword observations are corrupted by independent, equal-variance Gaussian noise, the inner-product maximization is equivalent to minimizing the Euclidean distance over the set of legal codewords, so this decision corresponds to the maximum-likelihood criterion. It remains an optimality in the statistical sense and cannot guarantee the correction of all multi-bit errors beyond the error-correction radius of the minimum distance.

4 Experiments

4.1 Experimental Setup

We evaluate on 400 videos generated by HunyuanVideo-I2V (Kong et al., 2024) from prompts sampled from the VBench-2.0 prompt set (Zheng et al., 2025), with watermark embedding and extraction in the model's 16-channel, 64×64 latent space. All videos share one generation configuration, kept identical across the no-watermark reference, the full method, and all baselines. We compare with three representative video diffusion watermarking baselines—VideoShield (Hu et al., 2025a), VideoMark (Hu et al., 2025b), and SIGMark (Zhao et al., 2026)—using the authors' public implementations under the unified attack protocol of WAVES (An et al., 2024): all methods use the same 400 videos, attack parameters, and 512-bit watermark length. Detailed prompt allocation, hyper-parameters, and baseline mechanisms are in Appendices B and D.

The primary metric is the 512-bit watermark bit accuracy

$$\mathrm{BitAcc} = 1 - \frac{d_H(\hat{m}, m)}{512}, \tag{6}$$

where d_H denotes the Hamming distance. We report the mean over the 400 videos. Imperceptibility is measured by the V-Score of VBench-2.0 (Zheng et al., 2025); its computation and the attack settings are in Appendix C.

Table 1: Bit accuracy (512-bit watermark) and V-Score of video diffusion watermarking methods under clean and three perturbed conditions.

Method	V-Score	Clean	Gaussian noise	H.264	Gaussian blur
VideoMark	**0.483**	0.7089	0.5137	0.5142	0.5391
VideoShield	0.482	**1.0000**	**0.9932**	0.8452	0.9705
SIGMark	0.472	0.9444	0.5048	0.4993	0.4945
SelfCheckMark (ours)	0.479	**1.0000**	0.9890	**0.8536**	**0.9804**

Table 2: Bit accuracy of SelfCheckMark at different watermark capacities under clean, Gaussian-noise, H.264, and Gaussian-blur conditions.

Capacity (bits)	Clean	Gaussian noise	H.264	Gaussian blur
512	**1.0000**	0.9890	0.8536	0.9804
256	1.0000	**1.0000**	0.9374	0.9993
128	1.0000	1.0000	0.9794	**1.0000**
64	1.0000	1.0000	0.9931	1.0000
32	1.0000	1.0000	**1.0000**	1.0000

4.2 Watermark Robustness and Imperceptibility

Table 1 compares the bit accuracy and V-Score of SelfCheckMark with the three baselines under the same videos, attack parameters, and watermark length.

SelfCheckMark reaches 100% under clean conditions, on par with the best baseline, and 98.90% under Gaussian noise, 0.42 points below VideoShield but above the rest. On H.264 and blur it attains 85.36% and 98.04%, improving over VideoShield by 0.84 and 0.99 points, the best average on the hardest perturbation, indicating that confidence fusion and codeword self-correction mitigate compression-induced inversion errors.

Table 2 reports the same method at lower watermark capacities under the same four conditions; bold marks, for each condition, the least redundant setting that already reaches 1.0000. The accuracy rises monotonically as the capacity decreases, from 85.36% to 97.94% under H.264 between 512 and 128 bits, and reaches 1.0000 under every condition at 32 bits.

For imperceptibility, since the watermark only changes the sign combination of the initial noise, per-pixel correspondence between watermarked and unwatermarked videos is not required; we therefore use the V-Score of VBench-2.0 (Zheng et al., 2025) rather than PSNR or SSIM (Wang et al., 2004). The second column of Table 1 reports the V-Score of each method. On the same videos, the unwatermarked reference scores 0.463, and all four methods fall in $[0.472, 0.483]$, within 0.020 of it, with SelfCheckMark at 0.479; such margins are far too small to reflect any perceptible change, so no method shows a substantive advantage or penalty. SelfCheckMark thus attains the highest recovery accuracy under H.264 and blur without trading video quality for robustness.

4.3 Capacity–Robustness Trade-off Analysis

The extended codeword has 1024 embedding slots, so capacity and redundancy trade off directly under a fixed slot budget. To separate the effects of "error-correcting code structure" and "total redundancy", we compare the extended Hamming(8,4) coding ("with Hamming") against a pure repetition scheme ("without Hamming") that soft-combines the same bit over multiple slots. At 512 bits both have code rate 1/2 and occupy identical slots; at 256, 128, 64, and 32 bits the slots per information bit of both rise synchronously. Results are in Figure 2; numerical values in Appendix E.

At 512 bits, pure repetition is comprehensively better: 0.47, 4.03, and 0.64 percentage points higher under noise, H.264, and blur (both stay at 1.0000 under clean), with the largest gap under H.264 (0.8536 vs. 0.8939). This stems from the intra-block correlation of errors: about 57% of 8-bit blocks under H.264 contain two or more errors, exceeding the single-error correction radius, so the

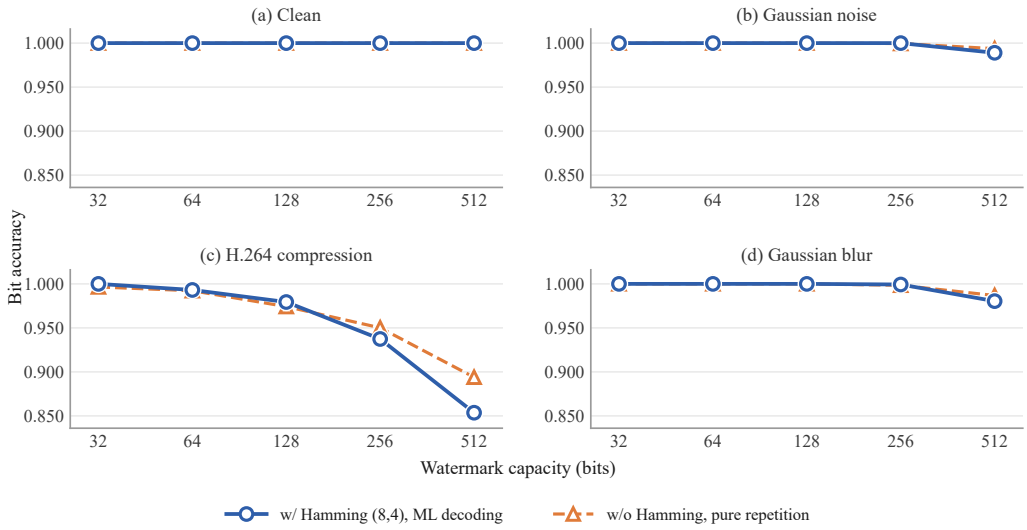

Figure 2: Capacity–robustness trade-off. Bit accuracy of the extended Hamming(8,4) coding (solid) and pure repetition (dashed) strategies as a function of watermark capacity under (a) clean, (b) Gaussian noise, (c) H.264 compression, and (d) Gaussian blur.

parity bits occupy slots that repetition could use more profitably; in contrast, $2\times$ repetition gains an SNR of $\sqrt{2}$ through soft combining, which is insensitive to the error correlation structure.

As capacity decreases the gap converges: the H.264 gap narrows to 1.27 points at 256 bits, and at 128 bits and below the ordering reverses, with Hamming ahead under H.264 (0.9794 vs. 0.9742 at 128 bits, 1.0000 vs. 0.9965 at 32 bits) and indistinguishable under noise and blur, where both saturate at 1.0000. Repetition therefore leads only where the slots per information bit are too few for the parity bits to be spent profitably. Thus the parity bits' marginal value rises with total redundancy, and all curves rise monotonically as capacity decreases, verifying the trade-off.

The reversal also matters because 128 bits is already sufficient for the intended use: retaining the field structure of Section 3.1, it carries the full 32-bit version field and a 96-bit custom field, i.e., $2^{96} \approx 7.9\times10^{28}$ identifiable keys, far exceeding the low-entropy identifiers needed for attribution and provenance, while its H.264 accuracy of 0.9794 keeps recovery essentially loss-free. At capacities of 128 bits or fewer, extended Hamming with maximum-likelihood decoding matches or surpasses pure repetition under the evaluated attacks. Because a 128-bit payload already supports the intended attribution task, we retain this coding scheme as the default for its interpretable codeword-level decision, while acknowledging its robustness cost at 256 and 512 bits.

4.4 Ablation Studies

The ablations target the two core modules: the two-level fusion, and the codeword-level self-correction. The former examines whether latent soft magnitudes and replica stability provide additional reliability information, and the latter whether joint soft decision over legal codewords beats per-bit hard decision. All configurations use the same 400 videos.

4.4.1 Two-Level Confidence Fusion

We decompose the two-level fusion into two factors—"local evidence form" (Sign-only vs. Soft amplitude) and "replica fusion scheme" (Mean vs. invstd)—forming four controlled configurations. Sign-only quantizes each block to signs and sums them, $q_{r,i}^{\text{sign}} = \sum_{p=1}^{4} \text{sign}(z'_{r,i,p})$ (a 2:2 tie yields invalid evidence); Soft amplitude sums the continuous descrambled values, so larger magnitudes contribute more. Mean averages the 256 replicas equally; invstd weights each replica by the inverse

Table 3: Effect of two-level confidence fusion on the final 512-bit watermark accuracy.

Local evidence	Replica fusion	Gaussian noise	H.264	Gaussian blur
Sign-only	Mean	0.9771	0.8088	0.9646
Soft amplitude	Mean	0.9883	0.8497	0.9786
Sign-only	invstd	0.9766	0.8086	0.9645
Soft amplitude	invstd	**0.9890**	**0.8536**	**0.9804**

Table 4: Bit accuracy of different watermark self-correction strategies.

Self-correction strategy	Gaussian noise	H.264	Gaussian blur
No self-correction	0.9618	0.8094	0.9460
Hard syndrome correction	0.9797	0.8279	0.9671
Soft-guided correction	0.9799	0.8334	0.9682
Maximum-likelihood (ours)	**0.9890**	**0.8536**	**0.9804**

standard deviation of its evidence, down-weighting fluctuating ones. All four configurations use the same Hamming maximum-likelihood decoder for the final codeword decision.

Fixing the replica fusion, preserving soft magnitudes improves accuracy by 1.12, 4.09, and 1.40 points under noise, H.264, and blur (Mean) and by 1.24, 4.50, and 1.59 points (invstd), confirming that latent magnitudes provide effective element-level reliability. The gain under H.264 (3–4$\times$ the others) reflects that compression creates many low-confidence observations that Sign-only weights equally but Soft amplitude down-weights.

Comparing the 1st and 3rd rows of Table 3, under Sign-only, changing Mean to invstd brings almost no gain (-0.05, -0.02, -0.01 points): once block elements are quantized to signs, the replica standard deviation no longer reflects reliability. Under Soft amplitude the same change gains 0.07, 0.39, and 0.18 points, so replica fluctuations are informative only when continuous magnitudes are preserved and the complete two-level fusion is best.

4.4.2 Hamming Watermark Self-Correction

We fix Soft amplitude local fusion and invstd replica fusion and vary only the final decoding scheme: *No self-correction* decides bit by bit from the sign of s; *Hard syndrome correction* binarizes s and corrects single-bit errors per 8-bit block via the standard Hamming table (blocks beyond the radius keep their decision); *Soft-guided correction* flips the lowest-confidence position first and checks the syndrome, still bit by bit; *Maximum-likelihood (ours)* compares the matching scores over all 16 legal codewords of each block.

Hard syndrome correction improves over no self-correction by 1.79, 1.85, and 2.11 points (noise, H.264, blur), verifying the benefit of structured redundancy; soft-guided correction adds only 0.02, 0.55, and 0.11 points, so threshold-style guidance still under-uses the continuous evidence. Maximum-likelihood decoding is best everywhere, improving over hard correction by 0.93, 2.57, and 1.33 points and over no self-correction by 2.72, 4.42, and 3.44 points, with the largest gain under H.264, where blocks often contain multiple unreliable positions.

5 Conclusion

Blind watermarking of generated videos is critical for copyright attribution and provenance tracing in AIGC. We propose SelfCheckMark, a confidence-driven watermarking and self-correcting extraction framework for video diffusion models. We build a sparse-anchor plus continuous-redundancy dual-channel embedding, fuse the latent soft magnitudes with inverse-standard-deviation replica weighting, and recover the 512-bit watermark information by maximum-likelihood soft decision over extended Hamming codewords. Experiments show that SelfCheckMark recovers the watermark reliably under Gaussian noise, H.264 compression, and Gaussian blur, attaining superior or competitive accuracy over the baselines without degrading video quality.

REPRODUCIBILITY STATEMENT

All experiments are conducted with the publicly available HunyuanVideo-I2V generation model and the VBench-2.0 prompt set, and every method (the full method, all baselines, and all ablation configurations) is evaluated on the same 400 videos with the same attack parameters and the same 512-bit watermark length. Fixed random seeds are used throughout prompt selection, video generation, watermark writing, and robustness perturbation, and the three-level data index (evaluation dimension–prompt index–generation index) is documented in Appendix B. The robustness attack settings (Gaussian noise, H.264 compression, and Gaussian blur) are specified in Appendix C, and the generator/parity-check matrices and the full production codebook of the extended Hamming(8,4) code are provided in Appendix A. The choice of video generation hyper-parameters (resolution, frame count, frame rate, timestep scheme, and guidance scale) is given in Appendix D. Anonymous source code will be released as supplementary material upon publication.

AI USE STATEMENT

In this work, we used generative AI tools for research ideation. We have not used generative AI tools for generating synthetic data sets or for proving mathematical claims. Additionally, we used generative AI tools to aid and polish the writing, to draft portions of the manuscript, and for retrieval and discovery, such as identifying related work. We have reviewed all AI-assisted work: AI-suggested references were verified against the original publications through a manual literature survey, and all AI-assisted text was checked for technical correctness and accuracy by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies watermarking for AI-generated video and does not involve human subjects. The proposed method is intended to support copyright attribution, provenance tracing, and platform governance of generated content, and we do not foresee direct negative ethical consequences from its use. All experiments use publicly available models and benchmark prompts.

REFERENCES

Bang An, Mucong Ding, Tahseen Rabbani, et al. Waves: Benchmarking the robustness of image watermarks. In *International Conference on Machine Learning (ICML)*, 2024.

David Chase. Class of algorithms for decoding block codes with channel measurement information. *IEEE Transactions on Information theory*, 18(1):170–182, 1972.

Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Annual International Cryptology Conference*, pp. 325–347. Springer, 2024.

Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.

Samar Fares, Nurbek Tastan, and Karthik Nandakumar. Spdmark: Selective parameter displacement for robust video watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10303–10312, 2026.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22409–22420. IEEE, 2023.

G Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12 (2):125–131, 1966.

Richard W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646, 2022.

Runyi Hu, Jie Zhang, Yiming Li, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Videoshield: Regulating diffusion-based video generation models via watermarking. In *International Conference on Learning Representations*, volume 2025, pp. 52395–52425, 2025a.

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. arXiv preprint arXiv:2504.16359, 2025b.

Huayang Huang, Yu Wu, and Qian Wang. Robin: Robust and invisible watermarks for diffusion models with adversarial optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 3937–3963, 2024a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024b.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models, 2024.

Shangran Lin, Lu Lu, Jian Chen, and Qiang Liu. Rapid: Reusing attention sparsity with inter-step adaptation for efficient video diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36147–36156, June 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xiyang Luo, Yinxiao Li, Huiwen Chang, Ce Liu, Peyman Milanfar, and Feng Yang. Dvmark: a deep multiscale framework for video watermarking. *IEEE Transactions on Image Processing*, 34:4371–4385, 2023.

Yoav Nir and Adam Langley. Chacha20 and poly1305 for ietf protocols. RFC 8439, 2018.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2114–2123. IEEE, 2020.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58047–58063, 2023.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, Peizhen Zhang, Peng Chen, Penghao Zhao, Qi Tian, Songtao Liu, Weijie Kong, Weiyan Wang, Xiao He, Xin Li, Xinchi Deng, Xuefei Zhe, Yang Li, Yanxin Long, Yuanbo Peng, Yue Wu, Yuhong Liu, Zhenyu Wang, Zuozhuo Dai, Bo Peng, Coopers Li, Gu Gong, Guojian Xiao, Jiahe Tian, Jiaxin Lin, Jie Liu, Jihong Zhang, Jiesong Lian, Kaihang Pan, Lei Wang, Lin Niu, Mingtao Chen, Mingyang Chen, Mingzhe Zheng, Miles Yang, Qiangqiang Hu, Qi Yang, Qiuyong Xiao, Runzhou Wu, Ryan Xu, Rui Yuan, Shanshan Sang, Shisheng Huang, Siruis Gong, Shuo Huang, Weiting Guo, Xiang Yuan, Xiaojia Chen,

Xiawei Hu, Wenzhi Sun, Xiele Wu, Xianshun Ren, Xiaoyan Yuan, Xiaoyue Mi, Yepeng Zhang, Yifu Sun, Yiting Lu, Yitong Li, You Huang, Yu Tang, Yixuan Li, Yuhang Deng, Yuan Zhou, Zhichao Hu, Zhiguang Liu, Zhihe Yang, Zilin Yang, Zhenzhi Lu, Zixiang Zhou, and Zhao Zhong. Hunyuanvideo 1.5 technical report, 2025. URL https://arxiv.org/abs/2511.18870.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12162–12171. IEEE, 2024.

Zijing Zhao, Hui Jin, Qingxiao Guo, Yilong Ma, YUNHAO WANG, Xiaobing Guo, Weifeng Zhang, et al. Sigmark: Scalable in-generation watermark with blind extraction for video diffusion. In *International Conference on Learning Representations*, volume 2026, pp. 150652–150665, 2026.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness, 2025.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European conference on computer vision*, pp. 682–697. Springer, 2018.

A Details of the Extended Hamming Coding

A.1 Payload Fields and Data Bit Order

The original 512-bit watermark is formed by concatenating 480-bit Custom information and 32-bit Version information in order. Custom information carries adaptive information for content, users, or application scenarios. In Version information, the first 16 bits indicate the video generation model and its version, the middle 8 bits indicate the VAE and inversion configuration version, and the last 8 bits indicate the watermark protocol version. The concatenated bit stream forms an information vector $u = [u_1, u_2, u_3, u_4]$ every 4 bits from left to right, and the codeword is computed in row-vector form as $c = u\,G \pmod 2$.

The generator matrix consistent with the production codebook is

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{7}$$

Under the codeword order $c = [c_1, c_2, \ldots, c_8]$, the data bits are located at c_3, c_5, c_6, c_7, corresponding to u_1, u_2, u_3, u_4, respectively. A parity-check matrix matching this generator matrix is

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad GH^\top = 0. \tag{8}$$

A.2 Production Codebook

Table 5 lists the complete mapping from 4-bit information blocks to the 8-bit extended Hamming codewords used in our implementation.

B Construction of the VBench-2.0 Evaluation Prompt Set

We sample from the VBench-2.0 prompt set enhanced for advanced video diffusion models such as HunyuanVideo, covering 18 evaluation dimensions: Human Anatomy, Human Clothes, Human Identity, Composition, Diversity, Mechanics, Material, Thermotics, Multi-View Consistency, Dynamic Spatial Relationship, Dynamic Attribute, Motion Order Understanding, Human Interaction, Complex Landscape, Complex Plot, Camera Motion, Motion Rationality, and Instance Preservation.

Table 5: Correspondence between 4-bit information blocks and 8-bit legal codewords.

Information	Codeword	Information	Codeword
0000	00000000	1000	11100001
0001	11010010	1001	00110011
0010	01010101	1010	10110100
0011	10000111	1011	01100110
0100	10011001	1100	01111000
0101	01001011	1101	10101010
0110	11001100	1110	00101101
0111	00011110	1111	11111111

Table 6: Allocation of evaluation prompts and video counts.

Dimension	Prompts	Videos per prompt	Total videos
Diversity	3	20	60
Remaining 17 dimensions	85 (5 each)	4	340
Total	88	–	400

Fixed random seeds are used for prompt selection, video generation, watermark writing, and robustness perturbations, to ensure that all methods and ablation configurations share consistent sample conditions; for experimental configuration management reasons, the paper does not disclose the specific seed values. The data index adopts a three-level relation of "evaluation dimension–prompt index–generation index": in Diversity, each prompt corresponds to 20 generation indices; in the other dimensions, each prompt corresponds to 4 generation indices. The baselines and SelfCheck-Mark read the same video and its attacked versions through the same data index, thereby ensuring per-sample correspondence and consistent comparison protocols.

C Robustness Attacks and Video Quality Metric Computation

C.1 Robustness Attacks

We evaluate three types of common video processing—Gaussian noise, H.264 compression, and Gaussian blur—which correspond to transmission noise, lossy coding, and spatial filtering, respectively, and are among the most commonly used perturbations in video watermarking evaluation (An et al., 2024; Hu et al., 2025a). All three attacks are applied to the watermarked video in the pixel domain, with the specific settings shown in Table 7.

Table 7: Robustness attack settings.

Condition	Setting
Clean	No additional perturbation
Gaussian noise	Additive noise with σ=0.05 after normalizing pixels to $[0, 1]$
H.264 compression	CRF=35, preset=medium, pixel format yuv420p
Gaussian blur	5×5 Gaussian kernel, σ=1.5

The selection of the attack parameters in Table 7 is consistent with related video watermarking works (Hu et al., 2025a) so that the results are directly comparable: H.264 compression uses a relatively aggressive compression strength of CRF=35 and preset=medium, with pixel format yuv420p; Gaussian noise adds additive noise with σ=0.05 after normalizing pixel values to $[0, 1]$; Gaussian blur uses a 5×5 convolution kernel with σ=1.5.

C.2 Video Quality Metric Computation

The imperceptibility of the watermark is measured by the video generation quality, using the V-Score of VBench-2.0 (Zheng et al., 2025) as the evaluation metric. The range of V-Score is $[0, 1]$, with higher values indicating better generation quality. The evaluation follows the official VBench-2.0 process, without modifying its dimension composition and aggregation weights; the no-watermark reference and each watermarking method generate videos under the same batch of prompts, the same random seeds, and completely identical generation configurations, so the differences in V-Score come only from the watermark embedding itself. We report the mean V-Score of all videos under the same conditions.

D Detailed Experimental Setup

D.1 Model and Dataset

The video generation model is HunyuanVideo-I2V (Kong et al., 2024), an image-conditioned video diffusion model. Watermark embedding and extraction are performed in its latent space with 16 channels and 64×64 spatial resolution, with each video corresponding to 16 watermarked latent frames. We choose this model because its latent space has moderate dimensionality and its inversion implementation is public, making it convenient to compare with existing video diffusion watermarking methods under the same conditions. The evaluation prompt set is built from the VBench-2.0 prompt set (Zheng et al., 2025), which extends the VBench video generation benchmark suite (Huang et al., 2024b), and is enhanced for advanced video diffusion models such as HunyuanVideo; the detailed prompt allocation and sampling rules are given in Appendix B.

D.2 Video Generation Hyper-parameters

The video resolution is 512×512, each video contains 65 frames, and the frame rate is 8 fps. The generation process uses the 50-step FlowMatch Euler timestep scheme (Lipman et al., 2023), with a guidance scale of 6.0; extraction also runs a 50-step FlowMatch Euler reverse process with an empty text prompt. These parameters are kept identical across the no-watermark reference, the full method, and all baselines, so that videos under different conditions differ only in the watermark embedding scheme.

D.3 Watermark Baselines

VideoShield (Hu et al., 2025a) maps watermark bits onto the initial-noise template of video diffusion, extracts the template through DDIM inversion, and supports spatial and temporal tampering localization; it is one of the most thoroughly evaluated in-generation video watermarking methods. VideoMark (Hu et al., 2025b) adopts frame-level pseudo-random error-correcting codes and adapts to temporal perturbations such as frame dropping through a temporal matching module. SIGMark (Zhao et al., 2026) introduces a global frame-level pseudo-random encoding key to achieve blind extraction without per-video auxiliary information. All three baselines use the authors' public implementations and are re-evaluated under the unified protocol described in the main text.

E Capacity–Robustness Trade-off Results

The full bit-accuracy values underlying Figure 2 are reported in Table 8. The results cover five watermark capacities and four video conditions, comparing extended Hamming coding with pure repetition under each setting.

Table 8: Bit accuracy (mean) of the extended Hamming(8,4) coding ("with Hamming") and the pure repetition ("without Hamming") strategies at different capacity levels.

Attack	Capacity (bits)	with Hamming	without Hamming
Clean	512	1.0000	1.0000
	256	1.0000	1.0000
	128	1.0000	1.0000
	64	1.0000	1.0000
	32	1.0000	1.0000
Gaussian noise	512	0.9890	**0.9937**
	256	**1.0000**	0.9995
	128	1.0000	1.0000
	64	1.0000	1.0000
	32	1.0000	1.0000
H.264	512	0.8536	**0.8939**
	256	0.9374	**0.9501**
	128	**0.9794**	0.9742
	64	**0.9931**	0.9922
	32	**1.0000**	0.9965
Gaussian blur	512	0.9804	**0.9868**
	256	**0.9993**	0.9984
	128	1.0000	1.0000
	64	1.0000	1.0000
	32	1.0000	1.0000