UNIFYING WATERMARKING VIA DIMENSION-AWARE MAPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep watermarking methods often share similar encoder–decoder architectures, yet differ substantially in their functional behaviors. We propose DiM, a new multi-dimensional watermarking framework that formulates watermarking as **a dimension-aware mapping problem**, thereby unifying existing watermarking methods at the functional level. Under DiM, watermark information is modeled as payloads of different dimensionalities, including one-dimensional binary messages, two-dimensional spatial masks, and three-dimensional spatiotemporal structures. We find that the dimensional configuration of embedding and extraction largely determines the resulting watermarking behavior. Same-dimensional mappings preserve payload structure and support fine-grained control, while cross-dimensional mappings enable spatial or spatiotemporal localization. We instantiate DiM in the video domain, where spatiotemporal representations enable a broader set of dimension mappings. Experiments demonstrate that varying only the embedding and extraction dimensions and their handling modules leads to different watermarking capabilities, including spatiotemporal tamper localization, local embedding control, and recovery of temporal order under frame disruptions.

1 INTRODUCTION

Deep watermarking is a fundamental technique for ensuring copyright attribution Zhang et al. (2021) and content integrity Neekhara et al. (2024) in visual media. In recent years, the end-to-end training paradigm has provided a unified and flexible framework for deep watermarking Zhu et al. (2018); Luo et al. (2020). By adopting encoder–decoder architectures and incorporating noise layers to simulate attack processes, a large body of work has evolved under this paradigm, achieving varying trade-offs between robustness and visual quality Jia et al. (2021); Wu et al. (2023); Tancik et al. (2020). Despite sharing highly similar network architectures, existing methods are typically designed for specific tasks. Most prior approaches focus on embedding and extracting one-dimensional binary messages for copyright verification Bui et al. (2025); Hu et al. (2025b). With the growing demand for content authentication and tamper detection, some methods introduce additional fragile watermarks to enable spatial localization Zhang et al. (2024a; 2025), while others formulate localization as a spatially resolved prediction problem at the decoding stage Hu et al. (2025c); Sander et al. (2025). As a result, different watermarking functionalities are realized through task-specific designs, often without a shared modeling perspective. This observation raises a natural question: *can these seemingly disparate watermarking designs be unified under a common framework?*

To answer this question, we propose **DiM**, a unified watermarking framework that formulates watermarking as *a dimension-aware mapping problem.* Within DiM, watermark information is modeled as payloads of different dimensionalities, including one-dimensional binary messages, two-dimensional spatial masks, and three-dimensional spatiotemporal structures (see Figure 1). Distinct dimensional configurations naturally correspond to different watermarking capabilities. Specifically, when embedding and extraction are performed in the same dimensional space, the watermark enables precise and fine-grained control (*e.g.*, copyright verification). When embedding occurs in a lower-dimensional space and extraction in a higher-dimensional space, the watermark naturally supports coarser-grained functionalities (*e.g.*, tamper localization). Conversely, employing high-dimensional embedding with low-dimensional extraction enables progressive decoupling and recovery of information, effectively reducing decoding complexity and ensuring stable recovery when temporal dependencies in the payload are weak.

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.

Under the unified DiM, existing image and video watermarking methods can be understood as specific instances of the proposed framework rather than isolated, task-driven designs. However, prior work on cross-dimensional mappings has been largely restricted to one- and two-dimensional settings. To study more general dimensional configurations, we focus on video watermarking and instantiate DiM as a concrete video watermarking approach, denoted as **DiM-V**. The inherently high-dimensional spatiotemporal structure of video enables a broader range of embedding and extraction dimension combinations, making it a suitable setting for examining their functional consequences.

Guided by DiM, we evaluate multiple dimensional settings and show that merely adjusting the embedding–extraction dimensionalities and the corresponding payload-handling modules leads to fundamentally different watermarking capabilities. We further verify these behaviors across different backbone architectures, demonstrating that the observed capabilities are induced by the dimensional mappings rather than tied to a specific network design. Beyond conventional copyright protection, DiM-V fills critical gaps in video watermarking by enabling spatiotemporal tamper localization and fine-grained local embedding. These findings establish DiM as both a unifying formulation and a practical design principle for dimension-aware watermarking. This dimension-aware view further suggests a natural way to protect temporal structure, a capability largely overlooked by existing video watermarking methods. Because video semantics depends on both frame content and temporal order, reordering visually intact frames can fundamentally alter the meaning of an event. We therefore use high-dimensional payloads to encode temporal information, enabling DiM-V to represent frame-order structure and recover the original sequence under frame-order disruptions.

2 Background

Our proposed framework is primarily grounded in two domains: image watermarking and video watermarking, which we briefly review from the perspective of *how watermark information is represented and extracted*.

Image Watermarking. Existing image watermarking methods Petrov et al. (2025); Hu et al. (2024); Bui et al. (2023); Wen et al. (2023) can be broadly categorized by how watermark information is embedded and extracted. The most common paradigm performs global embedding and extraction of a one-dimensional binary message, primarily targeting robust copyright verification under common distortions, as exemplified by Robust-Wide Hu et al. (2025b), StegaStamp Tancik et al. (2020), and MBRS Jia et al. (2021). To support tamper localization, some approaches retain global watermarking but embed additional fragile signals whose degradation reveals manipulated regions, such as EditGuard Zhang et al. (2024a) and OmniGuard Zhang et al. (2025). Another line of work achieves localization by reformulating extraction as a spatially resolved prediction problem. For example, WAM Sander et al. (2025) embeds a global one-dimensional message while performing pixel-wise extraction to identify tampered regions; MaskWM Hu et al. (2025c) further extends this paradigm with a mask-guided design, supporting global embedding with local extraction. Its MaskWM-ED variant additionally incorporates two-dimensional masks at the embedding stage to enable local embedding. These methods show that *different watermarking tasks require different information representations. Yet even for the same task, existing approaches adopt substantially different representations, often introduced ad hoc without a unified formulation.*

Video Watermarking. Compared to image watermarking, video watermarking Luo et al. (2023); Zhang et al. (2024b); Jiang et al. (2025); Souček et al. (2025); Ji et al. (2026); Ye et al. (2023) methods remain focused on global embedding. RivaGAN Zhang et al. (2019) employs attention mechanisms and adversarial training for robust video watermarking; REVMark Zhang et al. (2023) improves robustness to compression via temporal alignment and H.264-aware perturbations; VideoSeal Fernandez et al. (2024) extends image watermarking methods to videos through temporal watermark propagation. Despite these advances, current video watermarking methods suffer from two key limitations. *First,* watermark information is almost exclusively modeled as a global one-dimensional message, with little explicit modeling of spatial or temporal localization, thus restricting their capability for tasks such as tamper localization. *Second,* temporal order is insufficiently modeled, which constitutes a critical limitation for video watermarking applications Hu et al. (2025a). When frames are reordered, inserted, or shuffled, existing approaches struggle to encode or recover temporal structure.

Motivated by these observations, we propose a unified multi-dimensional perspective for watermark information modeling. This perspective organizes existing methods by dimensionality of the water-

mark representation and naturally exposes the limitations of current video watermarking designs, while enabling explicit modeling of spatial locality and temporal structure.

3 DiM: Unified Watermarking Framework

In this section, we formulate deep watermarking as a dimension-aware mapping problem. The central premise of DiM is that watermarking functionality is governed by the dimensionality with which watermark information is represented during embedding and recovered during extraction. Unlike conventional approaches that impose a predefined watermark form (*e.g.*, fixed-length one-dimensional bit sequences), we model watermark information as a random variable whose functional capacity is determined by the dimensionality of its representation.

3.1 Multi-dimensional Payload Space

We first define a set of multi-dimensional payload spaces, which serve as the representational foundation of DiM. Let $\mathcal{P}^{(d)}$ denote a watermark payload defined in a d-dimensional representation space. We consider three payload spaces corresponding to $d \in \{1, 2, 3\}$.

Definition 3.1 (1D Binary Payload)**.** The 1D payload, denoted as $\mathbf{p}^{(1)} \in \mathcal{P}^{(1)} = \{0, 1\}^L$, carries global, invariant binary information that identifies ownership or provenance, where L is the length of the binary payload.

Definition 3.2 (2D Spatial Payload)**.** The 2D payload $\mathbf{p}^{(2)} \in \mathcal{P}^{(2)} = \mathbb{R}^{H \times W \times C_p}$ represents spatially localized structural information that captures region-level or geometric constraints, where H, W are spatial dimensions and C_p denotes the number of payload channels.

Definition 3.3 (3D Spatiotemporal Payload)**.** The 3D payload $\mathbf{p}^{(3)} \in \mathcal{P}^{(3)} = \mathbb{R}^{T \times H \times W \times C_p}$ represents spatiotemporal structural information defined over space-time volumes, where T is the temporal dimension.

These payloads do not correspond to distinct tasks. Instead, they represent a unified information primitive instantiated across a spectrum of representational dimensionalities. Figure 1 illustrates the relationships between payload spaces and the corresponding same- and cross-dimensional mappings.

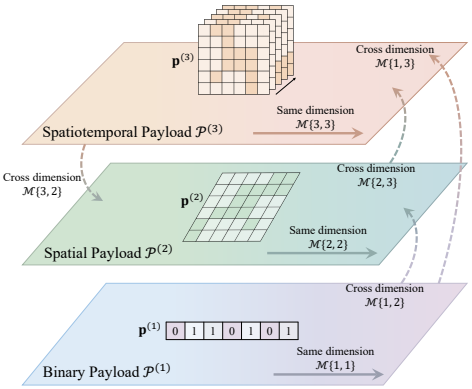

Figure 1: Dimension-aware embedding-extraction mappings across multi-dimensional watermark payloads. Watermark information is represented at different dimensionalities, and each mapping $\mathcal{M}\{d_e, d_d\}$ is defined by the dimensional relationship between embedding and extraction. Same-dimensional mappings operate within a single space, while cross-dimensional mappings bridge different dimensional spaces.

3.2 Dimension-aware Mapping

Let $x \in \mathcal{X}$ denote the host signal, such as an image or a video. We define the watermark embedding and extraction processes as dimension-aware functions:

$$\begin{aligned} \mathcal{E}_\theta &: \mathcal{X}_{\text{orig}} \times \mathcal{P}^{(d_e)} \to \mathcal{X}_{\text{wm}}, \\ \mathcal{D}_\phi &: \mathcal{X}_{\text{wm}} \to \hat{\mathcal{P}}^{(d_d)}, \end{aligned} \tag{1}$$

where $\mathcal{E}_\theta$ represents the embedding network (Encoder) with trainable parameters θ, taking the host x and a payload of dimension d_e as inputs. Similarly, $\mathcal{D}_\phi$ represents the extraction network (Decoder) with parameters ϕ, designed to recover a payload of dimension d_d. Let $\mathcal{A}$ denote an unknown distortion or attack process; the entire end-to-end pipeline can then be expressed as a unified mapping:

$$\mathcal{M}\{d_e, d_d\} : \mathcal{P}^{(d_e)} \xrightarrow{\mathcal{E}_\theta} \mathcal{X}_{\text{wm}} \xrightarrow{\mathcal{A}} \tilde{\mathcal{X}}_{\text{wm}} \xrightarrow{\mathcal{D}_\phi} \hat{\mathcal{P}}^{(d_d)}, \tag{2}$$

A key property of DiM is the relationship between the embedding dimension d_e and the extraction dimension d_d.

Table 1: Existing image and video watermarking methods under the DiM framework. Methods are grouped by embedding–extraction regimes $\mathcal{M}\{d_e, d_d\}$. Most prior work focuses on same-dimensional mappings, with limited coverage of cross-dimensional regimes.

$\mathcal{M}\{d_e, d_d\}$	$d_e = 1$	$d_e = 2$	$d_e = 3$
$d_d = 1$	Hidden, MaskWM-D, WAM, TrustMark, VideoSeal, REVMark, RivaGAN, EditGuard, OmniGuard, Robust-Wide, *etc*.	N/A	N/A
$d_d = 2$	MaskWM-D, WAM	MaskWM-ED, UDH, EditGuard, OmniGuard	N/A
$d_d = 3$	N/A	N/A	N/A

$\mathcal{M}\{d_e = d_d\}$: Same-dimensional Mapping. When embedding and extraction are performed in the same dimension, the extraction process is simplified to the estimation of the original payload. This regime preserves the payload structure, enabling fine-grained control (*e.g.*, copyright verification).

$\mathcal{M}\{d_e \neq d_d\}$: Cross-dimensional Mapping. When $d_e \neq d_d$, watermark extraction induces a *cross-dimensional projection*. We further distinguish two regimes.

- **Low-to-High Mapping $\mathcal{M}\{d_e < d_d\}$.** Low-dimensional payloads are embedded and recovered in a higher-dimensional representation space. This mapping expands a compact watermark signal into spatially or spatiotemporally resolved predictions, thereby enabling localization-oriented functionalities such as tamper detection.
- **High-to-Low Mapping $\mathcal{M}\{d_e > d_d\}$.** High-dimensional payloads are embedded, while extraction targets a lower-dimensional representation. Our empirical analysis indicates that this mapping becomes necessary when cross-dimensional coherence is weak, under which direct high-dimensional extraction is ill-conditioned.

From a functional perspective, DiM provides a unified view of existing image and video watermarking methods. Table 1 summarizes representative prior approaches under this framework, showing that most existing methods concentrate on the $\mathcal{M}\{1, 1\}$ regime, while several dimension combinations remain unexplored. Importantly, DiM is not intended as a taxonomy of existing watermarking methods, but as a functional design principle for realizing watermarking capabilities through dimension-aware payload mappings. Different choices of $\mathcal{M}\{d_e, d_d\}$ can thus produce distinct behaviors, including localized embedding and tamper localization. In the following section, we instantiate DiM in the video domain to to explore the N/A regimes and validate this design principle.

4 DiM-V: DiM for Video Watermarking

In this section, we instantiate DiM in the context of video watermarking, and refer to this concrete instantiation as DiM-V. *Our goal is not to introduce a task-specific video watermarking architecture, but to demonstrate how different dimension-aware mappings naturally give rise to diverse watermarking functionalities.*

4.1 Payload Instantiation

We consider a video clip $\mathbf{V}_{\text{orig}} \in \mathbb{R}^{T\times H\times W\times 3}$, where T denotes the temporal length and H, W denote the spatial resolution. Following the multi-dimensional payload formulation in Section 3.1, we instantiate three payload representations for video watermarking.

1D Binary Payload $\mathbf{W}$. The binary payload is instantiated as a randomly sampled binary message of length L, denoted as $\mathbf{W} \in \mathcal{P}^{(1)} = \{0, 1\}^L$. It encodes global, permutation-invariant information, such as ownership or identity, and is independent of the spatial or temporal structure of the video.

2D Spatial Payload $\mathbf{M}^{(2)}$. The spatial payload is instantiated as a spatial mask $\mathbf{M}^{(2)} \in \mathcal{P}^{(2)} = \mathbb{R}^{H\times W\times C_p^{\mathbf{M}^{(2)}}}$, which represents region-level structural information. Following MaskWM Hu et al. (2025c) and WAM Sander et al. (2025), we adopt the mask generation strategy of LaMa Suvorov et al. (2022), constructing four types of masks: full masks, rectangular masks, irregular masks, and

segmented masks. As segmented masks are defined on a per-frame basis and inherently indexed in time, we randomly sample a single frame-level mask and treat it as a static 2D spatial payload.

3D Spatiotemporal Payload $\mathbf{M}^{(3)}$. The 3D payload is instantiated as a spatiotemporal mask tensor $\mathbf{M}^{(3)} \in \mathcal{P}^{(3)} = \mathbb{R}^{T\times H\times W\times C_p^{\mathbf{M}^{(3)}}}$ which encodes temporally varying spatial structures across video frames. Consistent with the 2D setting, we consider the same four categories of masks. For rectangular and irregular masks, we employ a simple yet effective temporal mask-shifting strategy that propagates a single spatial mask across frames, thereby emulating object-tracking–like behavior along the temporal dimension (see Appendix D). In the specific case of full masks, setting $C_p^{\mathbf{M}^{(3)}} = 1$ results in identical masks across all frames and provides no temporal differentiation. More generally, conventional mask designs lack the capacity to encode or infer temporal order because they do not utilize an explicit, permutation-invariant representation of frame identity.

To address this limitation, we introduce a *multi-channel mask encoding mechanism* with $C_p^{\mathbf{M}^{(3)}} > 1$, where each frame is assigned a distinct permutation-invariant binary code along the channel dimension. All-zero codes are excluded to avoid ambiguity with the unmasked state. As illustrated in Figure 2, this representation enables frame-wise localization and reliable temporal order recovery under arbitrary frame permutations.

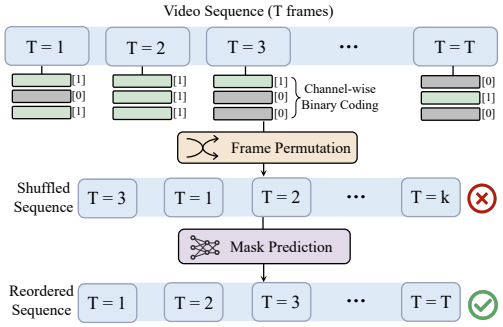

Figure 2: Multi-channel mask encoding for spatiotemporal payloads. Each frame receives a permutation-invariant binary channel code whose maps are spatially uniform (all zeros or ones), enabling frame-wise localization and temporal-order recovery under arbitrary frame permutations.

4.2 Input Construction

The input tensor $\mathbf{T}_{\text{in}} \in \mathbb{R}^{C_{\text{in}}\times T\times H\times W}$ is constructed by channel-wise concatenation of watermark-related features and the host video $\mathbf{V}_{\text{orig}} \in \mathbb{R}^{3\times T\times H\times W}$. The watermark-related features consist of two parts: a feature representation derived from the 1D message $\mathbf{W}$, and an optional spatial or spatiotemporal mask payload, depending on the mapping. The 1D message $\mathbf{W}$ is always included, as it serves as the core carrier of watermark identity and ensures copyright verification across all mappings. A 2D spatial mask $\mathbf{M}^{(2)}$ or a 3D spatiotemporal mask $\mathbf{M}^{(3)}$ can be optionally incorporated to induce locality or temporal structure. Below, we describe how $\mathbf{T}_{\text{in}}$ is instantiated under representative mappings and analyze the functional behaviors induced by each dimensional configuration.

Same Dimension $\mathcal{M}\{3,3\}$. 1D message $\mathbf{W}$ is first transformed by a message translator into a spatiotemporal feature tensor $\mathbf{T}_{\text{msg}} \in \mathbb{R}^{C_{\text{tp}}\times T\times H\times W}$, which is then concatenated with a 3D mask $\mathbf{M}^{(3)}$ and the original video $\mathbf{V}_{\text{orig}}$ along the channel dimension, yielding $\mathbf{T}_{\text{in}} = \text{Concat}\big(\mathbf{V}_{\text{orig}},\ \mathbf{T}_{\text{msg}},\ \mathbf{M}^{(3)}\big)$. This paradigm supports both global and local embedding of the message. Global embedding refers to embedding the payload across the entire video, while local embedding corresponds to embedding the payload within a specific object or region of interest, such as along an object trajectory.

Cross Dimension $\mathcal{M}\{1,3\}$. In this setting, the input tensor is constructed as $\mathbf{T}_{\text{in}} = \text{Concat}\big(\mathbf{V}_{\text{orig}},\ \mathbf{T}_{\text{msg}}\big)$. Embedding a low-dimensional payload while extracting a spatiotemporal representation induces a structure-expanding projection, enabling effective localization of tampering events such as frame removal or object deletion.

Cross-Dimension $\mathcal{M}\{2,3\}$. A 2D payload $\mathbf{M}^{(2)}$ is replicated along the temporal dimension to form a mask sequence $\mathbf{M}^{(3)}$ aligned with the video length. The input tensor is given by $\mathbf{T}_{\text{in}} = \text{Concat}\big(\mathbf{V}_{\text{orig}},\ \mathbf{T}_{\text{msg}},\ \mathbf{M}^{(3)}\big)$. Compared to $\mathcal{M}\{1,3\}$, this mapping introduces spatial constraints, enabling controlled local embedding while retaining spatiotemporal localization capability.

Cross Dimension $\mathcal{M}\{3,2\}$. The input tensor is the same as $\mathcal{M}\{3,3\}$. However, direct 3D extraction becomes ill-conditioned when high-dimensional spatiotemporal payloads have weak cross-frame

Table 2: Comparison of imperceptibility and robustness against baselines in terms of global watermarking in video scenarios. The symbols ❍ and ❑ indicate image and video watermarking methods, respectively. Bits denotes the number of embedded bits. Checkmarks indicate qualitative capabilities: L-E, L-X, and Loc denote local embedding, local extraction, and tamper localization, respectively. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Bits	PSNR ↑	SSIM ↑	Clean ↑	Common Distortions ↑		Video Distortions ↑		Capabilities		
					Valuemetric	Geometric	Frame-level	Compression	L-E	L-X	Loc
Same-dimensional Mapping $\mathcal{M}\{d_e = d_d\}$											
❍ MaskWM-ED	64	39.35	0.9704	**100**	**100**	99.92	96.88	54.47	✓	✓	✓
❍ OmniGuard	100	39.17	0.9759	99.98	92.68	71.13	96.89	91.71	✗	✗	✓
❍ TrustMark	100	40.92	0.9865	99.96	95.94	78.47	97.09	89.88	✗	✗	✗
❍ Robust-Wide	64	41.82	0.9904	**100**	99.14	50.42	96.88	**99.91**	✗	✗	✗
❑ VideoSeal	96	**53.86**	**0.9988**	98.54	81.39	97.28	98.68	54.12	✗	✗	✗
❑ RivaGAN	32	40.54	0.9793	99.82	95.95	83.06	97.29	70.27	✗	✗	✗
❑ REVMark	96	37.79	0.9905	99.97	96.69	57.25	96.63	88.22	✗	✗	✗
❑ **DiM-V** $\mathcal{M}\{3,3\}$	64	43.17	0.9902	**100**	**100**	**100**	**100**	98.58	✓	✓	✓
	96	42.36	0.9855	**100**	**100**	99.73	**100**	90.11	✓	✓	✓
	128	41.61	0.9847	**100**	**100**	98.91	**100**	92.99	✓	✓	✓
Cross-dimensional Mapping $\mathcal{M}\{d_e < d_d\}$											
❍ WAM	32	37.18	0.9643	**100**	99.90	99.43	96.88	71.42	✗	✓	✓
❍ MaskWM-D	64	38.84	0.9696	**100**	**100**	99.96	96.88	60.95	✗	✓	✓
❑ **DiM-V** $\mathcal{M}\{1,3\}$	64	43.88	0.9919	**100**	**100**	**100**	**100**	98.04	✗	✓	✓
❑ **DiM-V** $\mathcal{M}\{2,3\}$	64	43.60	0.9905	**100**	**100**	**100**	**100**	92.01	✓	✓	✓

coherence, especially as $C_p^{\mathbf{M}^{(3)}}$ increases. We therefore decouple mask prediction into frame-wise 2D extraction, which reduces prediction complexity and improves recovery stability and accuracy.

Regarding the outputs, $\mathcal{M}\{3,3\}$, $\mathcal{M}\{1,3\}$, and $\mathcal{M}\{2,3\}$ produce a 1D message together with a video-aligned 3D mask. In contrast, $\mathcal{M}\{3,2\}$ yields a 1D message and a sequence of frame-wise 2D masks aligned with the temporal dimension of the input video.

4.3 Watermarking Pipeline

Across all mappings, we employ a general network architecture and training pipeline equipped with a shared noise layer. Specifically, we adapt the MaskWM framework to the spatiotemporal domain and structure the pipeline into four stages: watermark embedding, watermark masking, watermark extraction, and mask prediction. Architectural and training details are provided in Appendix B. To further demonstrate the generality of DiM, we also instantiate it on the REVMark backbone, with detailed results provided in Appendix E.1. Experimental results show that, under the guidance of DiM, REVMark gains additional capabilities, including local embedding and tamper localization, without compromising its original robustness.

5 Experiments

5.1 Experimental Setup

Datasets. For all mapping configurations, we train our models on over 12,000 videos from the SA-V dataset Ravi et al. (2025). Training details are provided in Appendix A.1. Evaluation uses two subsets. The *global* subset contains 1,000 randomly sampled videos. The *local* subset is stratified into ten bins according to the ratio of masked area to the full video volume (0–10%, …, 90–100%), with 500 videos randomly sampled per bin. All videos have a resolution of 256×256 with $T = 8$ frames. Additional experiments on 720p and 1080p videos with 100+ frames are provided in Appendix C, where DiM preserves high imperceptibility, near-perfect robustness, and efficiency without retraining.

Metrics. Visual quality is assessed by PSNR and SSIM. Robustness is measured by Bit Accuracy under four categories of distortions: valuemetric, geometric, frame-level, and video compression. Valuemetric distortions include Gaussian noise, Gaussian blur, salt-and-pepper noise, and median

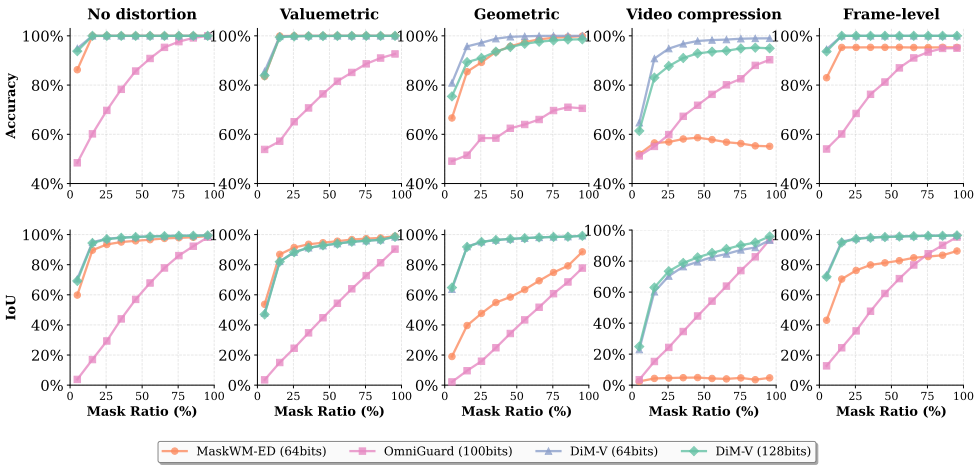

Figure 3: Comparison with baseline methods on local extraction (top) and tamper localization (bottom) in *the same-dimension mapping* setting. Results cover five distortion settings. Each mask-ratio point reports the mean over its interval (e.g., 5% denotes 1–10%).

filtering. Geometric distortions include rotation, perspective transformation, and horizontal flipping. Frame-level distortions include frame replacement, frame dropping, frame insertion, and temporal shuffling. Video compression is evaluated using H.264. Results are averaged across distortions within each category. Localization performance is measured by Intersection-over-Union (IoU) between predicted and ground-truth (gt) watermark regions. For the multi-channel setting, we report the mean IoU (mIoU) by averaging masks with distinct binary codes. Detailed distortion settings are provided in Appendix A.2.2. Efficiency are reported in Appendix E.3.

Baselines. We compare with nine recent open-source watermarking methods. Video watermarking baselines include VideoSeal Fernandez et al. (2024), Rivagan Zhang et al. (2019), and RevMark Zhang et al. (2023). Image watermarking baselines include MaskWM-D, MaskWM-ED Hu et al. (2025c), WAM Sander et al. (2025), OmniGuard Zhang et al. (2025), TrustMark Bui et al. (2025), and Robust-Wide Hu et al. (2025b). Image-based methods are evaluated using a frame-wise protocol inspired by VideoSeal, embedding the watermark into each frame and performing frame-level extraction and evaluation, with results averaged to obtain video-level performance. For methods that do not support the target resolution, we adopt the scaling strategy of TrustMark to match our setup.

5.2 Same-dimensional Mapping: $\mathcal{M}\{d_e = d_d\}$

Global 1D Embedding and Extraction. Table 2 compares imperceptibility, robustness, and functional capabilities under the same-dimensional mappings. Regardless of the embedding capacity, DiM-V demonstrates superior robustness against valuemetric, geometric, and frame-level distortions. Although Robust-Wide shows a slight advantage under compression due to its editing-oriented optimization, DiM-V achieves a favorable balance between robustness and imperceptibility. We also discuss strategies for improving compression robustness in Appendix E.2, where notable gains are observed. Beyond quantitative performance, DiM-V uniquely supports a combination of local embedding, local extraction, and tamper localization within a single configuration. Such functionality is largely absent from prior methods operating under the same-dimensional mapping regime.

Localized Extraction and Localization Performance. To assess these capabilities in the same-dimensional mapping, we compare against MaskWM-ED and OmniGuard, which support the $\mathcal{M}\{2, 2\}$ regime. As shown in Figure 3, DiM-V consistently achieves higher local extraction accuracy across all mask ratios and payload sizes. The performance gap becomes more pronounced under geometric and video-specific distortions, where baseline methods exhibit unstable extraction. With respect to localization performance, DiM-V remains robust across most distortions, with only a modest IoU decrease under valuemetric noise, representing trade-off given the overall robustness and functionality achieved.

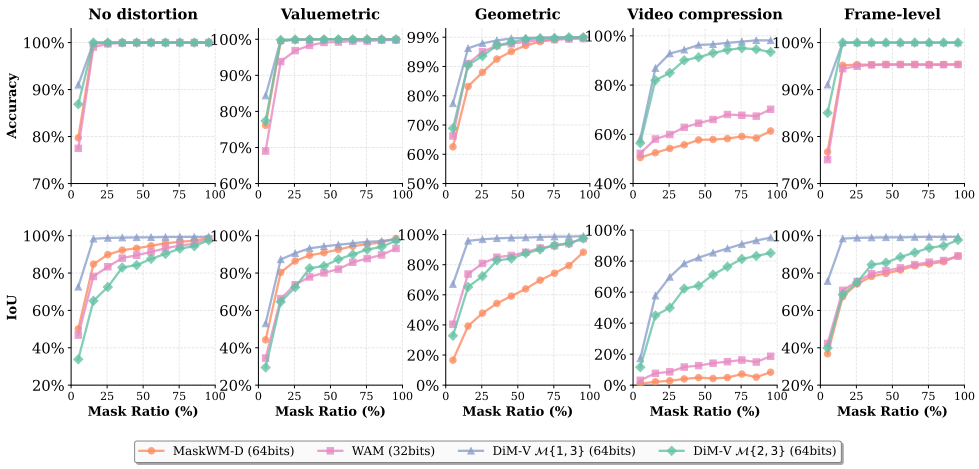

Figure 4: Comparison with baseline methods on local extraction (top) and tamper localization (bottom) in the *cross-dimension mapping* setting. Results cover five distortion settings. Each mask-ratio point reports the mean over its interval (e.g., 5% denotes 1–10%).

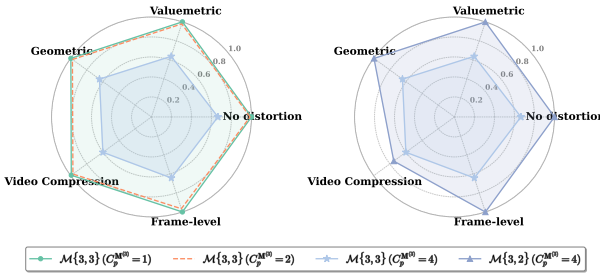

Figure 5: Effect of mask channel count $C_p^{\mathbf{M}^{(3)}}$ and dimensional mapping on global mIoU under diverse distortions.

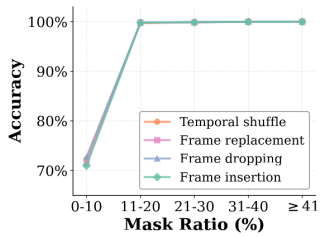

Figure 6: Bit accuracy (%) for multi-channel temporal mask recovery under varying mask ratios.

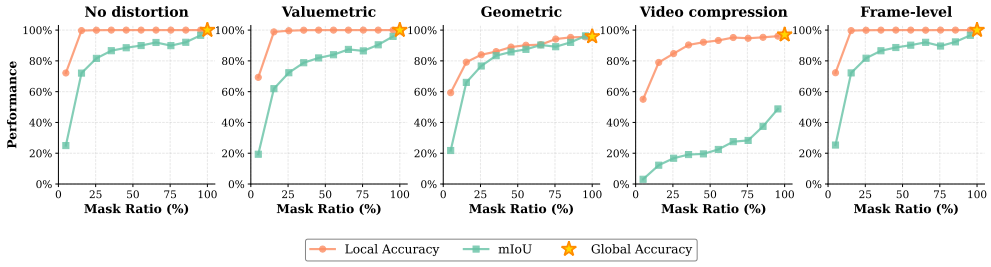

Figure 7: Evaluation of global and local watermark extraction accuracy alongside tamper localization performance under the *multi-channel mask design*. Results cover five distortion settings. Each mask-ratio point reports the mean over its interval (e.g., 5% denotes 1–10%).

5.3 Low-to-High Mapping: $\mathcal{M}\{d_e < d_d\}$

Global 1D Embedding and Extraction. As shown in Table 2, under the cross-dimensional mapping setting, our method significantly outperforms baselines, improving compression robustness by over 20% while ensuring accurate extraction across other distortion types. Moreover, we demonstrate significantly higher imperceptibility than WAM and MarkWM, realizing an optimal trade-off between robustness and visual fidelity.

Localized Extraction and Localization Performance. To evaluate low-to-high mappings, we compare DiM-V under $\mathcal{M}\{1,3\}$ and $\mathcal{M}\{2,3\}$ with MaskWM-D and WAM. As shown in Figure 4, $\mathcal{M}\{1,3\}$ achieves strong localization and extraction robustness across distortions. While $\mathcal{M}\{2,3\}$

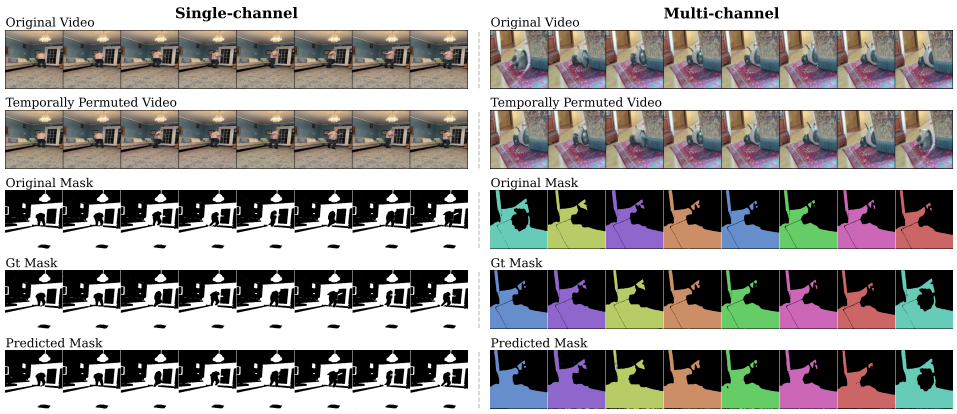

Figure 8: Visualization of single-channel and multi-channel mask representations under temporal permutation. Both methods enable spatial tamper localization. The multi-channel design assigns distinct encoded masks to different frames, visualized using different colors, allowing frame-specific identity inference and implicitly supporting temporal order localization.

performs slightly worse under clean and geometric conditions due to spatial mismatch between fixed embedding regions and dynamic tampering, it remains notably robust to video compression and frame-level distortions. These findings illustrate how dimensional expansion facilitates localization functionalities that significantly surpass those achievable by one-dimensional baselines.

5.4 High-to-Low Mapping: $\mathcal{M}\{d_e > d_d\}$

Necessity of Decoupled Extraction. Figure 5 analyzes the effect of multi-channel spatiotemporal payloads. In $\mathcal{M}\{3,3\}$, performance drops significantly in higher channel dimensions. This behavior reflects that weakened cross-frame coherence restricts the model's capacity to resolve the full video volume. In contrast, decoupling extraction into frame-wise prediction under $\mathcal{M}\{3,2\}$ restores localization performance, demonstrating the necessity of high-to-low projection in this regime.

Robustness against Frame-level Distortions. We evaluate $\mathcal{M}\{3,2\}$ for multi-channel temporal-mask recovery under frame-level distortions. Bit accuracy compares the embedded and extracted 32-bit codes (4 channels $\times$ 8 frames), with frame-level predictions obtained by majority voting within the predicted mask region. As shown in Figure 6, once the mask area exceeds 10%, the code is almost perfectly recovered across all distortions, demonstrating reliable temporal identity recovery. Performance degrades at 0–10%, where small regions hinder joint localization and identity-code preservation. Appendix C confirms scalability to higher-resolution and longer videos.

Impact of Multi-channel Mask Encoding. With four-channel mask encoding, DiM-V achieves a PSNR of 39.44 dB and an SSIM of 0.9807, outperforming most baselines despite a slight drop from single-channel encoding. As shown in Figure 7, DiM-V remains robust under most distortions, with moderate degradation under severe geometric transformations and compression due to more difficult multi-channel prediction. Tamper localization degrades for extremely small mask ratios, as precise localization of minute regions becomes harder when temporal indices are also encoded. Finally, Figure 8 shows that the encoded temporal indices preserve frame identity under temporal permutation, enabling frame-order recovery beyond spatial tamper detection.

6 Conclusion

We presented DiM, a dimension-aware framework that unifies deep watermarking by modeling embedding and extraction as mappings between payloads of different dimensionalities. By instantiating DiM in the video domain as DiM-V, we demonstrate that diverse watermarking behaviors, including local embedding, spatiotemporal tamper localization, and temporal order recovery, emerge naturally from different dimensional configurations. These capabilities are achieved without architectural modifications, highlighting representation dimensionality as a central factor governing watermarking functionality.

AI USE STATEMENT

Generative AI tools were used to improve the readability and presentation of this manuscript, including language polishing and formatting assistance. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content.

REFERENCES

Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 933–942, 2023.

Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Robust watermarking and watermark removal for arbitrary resolution images. In *IEEE International Conference on Computer Vision (ICCV)*, 2025.

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

Runyi Hu, Jie Zhang, Yiming Li, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Supermark: Robust and training-free image watermarking via diffusion-based super-resolution. *arXiv preprint arXiv:2412.10049*, 2024.

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 (ICLR)*, 2025a.

Runyi Hu, Jie Zhang, Ting Xu, Jiwei Li, and Tianwei Zhang. Robust-wide: Robust watermarking against instruction-driven image editing. In *European Conference on Computer Vision*, 2025b.

Runyi Hu, Jie Zhang, Shiqian Zhao, Nils Lukas, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Mask image watermarking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c.

Jianbin Ji, Dawen Xu, Li Dong, Lin Yang, and Songhan He. Dinvmark: A deep invertible network for video watermarking. *IEEE Transactions on Multimedia*, 2026.

Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, 2021.

Zhengyuan Jiang, Moyang Guo, Kecen Li, Yuepeng Hu, Yupu Wang, Zhicong Huang, Cheng Hong, and Neil Zhenqiang Gong. Videomarkbench: Benchmarking robustness of video watermarking. *arXiv preprint arXiv:2505.21620*, 2025.

Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13548–13557, 2020.

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

Paarth Neekhara, Shehzeen Hussain, Xinqiao Zhang, Ke Huang, Julian McAuley, and Farinaz Koushanfar. Facesigns: Semi-fragile watermarks for media authentication. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.

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

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations (ICLR)*, 2025.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel seal: Adversarial-only training for invisible image and video watermarking. *arXiv preprint arXiv:2512.16874*, 2025.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

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.

Xiaoshuai Wu, Xin Liao, and Bo Ou. Sepmark: Deep separable watermarking for unified source tracing and deepfake detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

Guanhui Ye, Jiashi Gao, Yuchen Wang, Liyan Song, and Xuetao Wei. Itov: efficiently adapting deep learning-based image watermarking to video watermarking. In *2023 International Conference on Culture-Oriented Science and Technology (CoST)*, pp. 192–197. IEEE, 2023.

Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 2020.

Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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

Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11964–11974, 2024a.

Xuanyu Zhang, Youmin Xu, Runyi Li, Jiwen Yu, Weiqi Li, Zhipei Xu, and Jian Zhang. V2a-mark: Versatile deep visual-audio watermarking for manipulation localization and copyright protection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024b.

Xuanyu Zhang, Zecheng Tang, Zhipei Xu, Runyi Li, Youmin Xu, Bin Chen, Feng Gao, and Jian Zhang. Omniguard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Yulin Zhang, Jiangqun Ni, Wenkang Su, and Xin Liao. A novel deep video watermarking framework with enhanced robustness to h. 264/avc compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

A MORE DETAILS

A.1 TRAINING DETAILS.

Under the single-channel mask setting, all videos are resized to 256×256 with 8 frames during training. Under the multi-channel mask setting, we train on videos of 128×128 with 8 frames. This reduced spatial resolution is adopted to improve training efficiency and memory usage. All models are trained on a single NVIDIA H100 GPU with a batch size of 8 for 200k steps. We use the AdamW optimizer with a learning rate of 2×10^{-4}, together with a cosine learning rate scheduler and 2,000 warm-up steps. Following MaskWM, we adopt a curriculum learning strategy from easy to hard. During the first 1,000 steps, only full-one masks are used and no distortions are applied. From 1,000 to 2,000 steps, all mask types are introduced. After 2,000 steps, distortion layers are enabled. The encoder loss weight β_{enc} is fixed to 1. The decoder loss weight β_{dec} is initialized to 20 and linearly decayed to 0.2 over the first 10k steps. The mask loss weight is set to 0.5. The JND module in the encoder is activated and optimized from 10k steps, with the scaling factor set to 1.

For VAE-based fine-tuning, VAE distortions are applied with 50% probability, while the remaining 50% use the original distortion types. The hyperparameters are set as follows: $\beta_{\text{enc}} = 0.3$, learning rate 2×10^{-4}, and the quality levels of *Bmshj18* and *Cheng20* are both set to 5.

A.2 DISTORTION DETAILS

A.2.1 TRAINING

Valuemetric Distortions. During training, valuemetric robustness is enhanced by randomly sampling from four common distortions: Gaussian blur, Gaussian noise, median filtering, and salt-and-pepper noise. The parameters are set as follows:

- **Gaussian blur:** kernel size = 1, standard deviation = 5.
- **Gaussian noise:** mean = 0, standard deviation = 0.1.
- **Median filter:** kernel size = 5.
- **Salt-and-pepper noise:** noise ratio = 0.1.

Geometric Distortions. To improve geometric robustness, we randomly sample from three typical geometric transformations: rotation, perspective transformation, and horizontal flipping. The configurations are as follows:

- **Rotation:** angle sampled from $[-90°, 90°]$.
- **Perspective:** distortion scale sampled from $[0.1, 0.5]$.
- **Horizontal flip:** no parameters.

Video Compression. We additionally incorporate an H.264-like compression distortion during training. Specifically, we adopt the distortion layer proposed in REVMark Zhang et al. (2023), which simulates both intra-frame and inter-frame compression effects. During training, the intra-frame compression strength is randomly sampled from $[1.5, 5]$, while the inter-frame compression strength is randomly sampled from $[5, 8]$.

Frame-level Distortions. To improve robustness to frame-related perturbations, we further apply the following distortions during training:

- **Frame shuffling:** randomly permute the frame order, with no constraint on the number of affected frames; in the most severe case, all frames are fully shuffled.
- **Frame replacement:** randomly replace one frame with an all-white frame (simulating frame substitution).
- **Frame dropping:** randomly drop one frame and append a new frame at the end (simulating frame loss).
- **Frame insertion:** randomly insert an all-white frame and remove the last frame (simulating insertion of an unrelated frame).

A.2.2 EVALUATION

Valuemetric Distortions. We apply four valuemetric distortions and evaluate robustness using the following parameter settings:

- **JPEG compression:** quality factor = 60.
- **Gaussian blur:** kernel size = 1, standard deviation = 3.
- **Gaussian noise:** mean = 0, standard deviation = 0.05.
- **Median filter:** kernel size = 3.
- **Salt-and-pepper noise:** noise ratio = 0.05.

Geometric Distortions. We apply three geometric distortions and evaluate robustness using the following configurations:

- **Rotation:** angle sampled from $[-30°, 30°]$.
- **Perspective:** distortion scale sampled from $[0.1, 0.3]$.
- **Horizontal flip:** no parameters.

Video Compression. We apply the built-in H.264 compression provided by the `torchvision.io` library when saving videos. Videos are encoded using the H.264 codec with a constant rate factor (CRF), where we evaluate two compression levels with CRF = 20 and CRF = 25. Lower CRF values correspond to higher visual quality and weaker compression, while higher CRF values indicate stronger compression.

Frame-level Distortions. We further evaluate robustness against frame-related perturbations using the same set of operations as in training, including:

- **Frame shuffling:** randomly permuting the frame order, with no constraint on the number of affected frames.
- **Frame replacement:** randomly replacing one frame with an all-white frame.
- **Frame dropping:** randomly dropping one frame and appending a new all-white frame at the end.
- **Frame insertion:** randomly inserting an all-white frame and removing the last frame.

B TRAINING

B.1 GENERAL WATERMARK PIPELINE

The encoder first processes the concatenated input and produces an intermediate encoded video $\mathbf{V}_{\text{enc}}$. To enhance perceptual quality, we apply a Just-Noticeable Difference (JND) module that adaptively modulates the embedding signal according to human visual sensitivity, resulting in the final watermarked video $\mathbf{V}_{\text{wm}}$.

$$\mathbf{V}_{\text{wm}} = \mathbf{V}_{\text{orig}} + \mu \times \text{JND}(\mathbf{V}_{\text{orig}}) \times (\mathbf{V}_{\text{enc}} - \mathbf{V}_{\text{orig}}) \tag{3}$$

To endow the model with mask-controlled embedding and extraction capabilities, as well as precise localization of manipulated regions, we adopt the MaskWM fusion strategy:

$$\mathbf{V}_{\text{fuse}} = \mathbf{V}_{\text{wm}} \odot \mathbf{M}^{(3)} + \mathbf{V}_{\text{orig}} \odot (1 - \mathbf{M}^{(3)}), \tag{4}$$

Masked regions preserve the embedded watermark, while unmasked regions are replaced by the original video content. The fused video $\mathbf{V}_{\text{fuse}}$ is then passed through a noise layer $\mathcal{A}$, which models a stochastic attack channel by randomly sampling transformations from a predefined distortion pool (see Appendix A.1 for details). The resulting video $\mathbf{V}'_{\text{fusion}}$ captures a wide range of appearance, geometric, and temporal degradations commonly encountered in practical scenarios.

$$\mathbf{V}'_{\text{fusion}} = \mathcal{A}(\mathbf{V}_{\text{fuse}}). \tag{5}$$

Finally, the masked extraction input is constructed as

$$\mathbf{V}_{\text{mask}} = \mathbf{V}'_{\text{fusion}} \odot \mathbf{M}, \tag{6}$$

which isolates watermark-bearing regions for subsequent watermark recovery $\mathbf{W}_{\text{pd}}$ in decoder, while the fused video $\mathbf{V}'_{\text{fusion}}$ is simultaneously used by the mask prediction network to infer the embedded mask $\mathbf{M}^{(3)}_{\text{pd}}$.

B.2 Network Architectures

Message Translator. A linear layer first maps $\mathbf{p}^{(1)}$ to a latent tensor of shape $(1,\, L,\, L,\, L \times (H/T))$, which is then resized via trilinear interpolation to $(1,\, T,\, H,\, W)$. A lightweight 3D CNN composed of multiple Conv–Norm–ReLU (CNR) blocks transforms this tensor into the final message representation $\mathbf{T}_{\text{msg}} \in \mathbb{R}^{C_{tp} \times T \times H \times W}$.

Encoder and Decoder. The encoder and decoder are adapted from MaskWM by extending all 2D convolutional modules to 3D convolutions, enabling direct operation on spatiotemporal video volumes. The encoder embeds the message and mask into the input video to produce a watermarked video, while the decoder recovers the 1D binary payload from $\mathbf{V}_{\text{mask}}$.

Mask Prediction Networks. To support both spatial and spatiotemporal payload extraction, we employ two mask prediction networks. For 2D spatial payload prediction in $\mathcal{M}\{3, 2\}$, we adopt the same U^2-Net architecture as MaskWM, which effectively captures fine-grained spatial structures. For 3D spatiotemporal payloads prediction in $\mathcal{M}\{3, 3\}$, $\mathcal{M}\{1, 3\}$ and $\mathcal{M}\{2, 3\}$, we extend U^2-Net by replacing all 2D convolutional modules with 3D counterparts, preserving its hierarchical multi-scale design while enabling temporal modeling.

Training Objectives. We follow the loss design of MaskWM. The overall training objective is defined as

$$\mathcal{L}_{\text{total}} = \beta_{\text{enc}} \mathcal{L}_{\text{enc}} + \beta_{\text{dec}} \mathcal{L}_{\text{dec}}, \tag{7}$$

where β_{enc} and β_{dec} balance the encoder and decoder losses, respectively. The encoder and decoder losses are defined as

$$\mathcal{L}_{\text{enc}} = \mathcal{L}_{\text{MSE}}(\mathbf{V}_{\text{wm}},\ \mathbf{V}_{\text{orig}})\,, \tag{8}$$

$$\mathcal{L}_{\text{dec}} = \mathcal{L}_{\text{MSE}}(\mathbf{W}_{\text{pd}},\ \mathbf{W}) + \alpha\, \mathcal{L}_{\text{MSE}}\Big(\mathbf{M}^{(3)}_{\text{pd}},\ \mathbf{M}^{(3)}\Big)\,, \tag{9}$$

where α controls the weight of the mask loss.

C High-resolution Long-video Evaluation

In this section, we further evaluate the scalability of DiM on long high-resolution videos. Specifically, we conduct additional experiments on 720p and 1080p videos with more than 100 frames. No retraining is required for this setting. Following the resolution-adaptive design of TrustMark Bui et al. (2025), each video is first segmented into 8-frame clips. For each clip, frames are downsampled to 256×256 for watermark embedding, and the predicted residual is then upsampled back to the original resolution and added to the input video. After embedding, the full watermarked video is attacked as a whole, and only then segmented again for watermark extraction. This protocol ensures that temporal attacks and video-level distortions are applied to the complete video rather than to isolated short clips.

As shown in Table 3, DiM maintains high imperceptibility and near-perfect robustness at both resolutions. Meanwhile, the system remains efficient, achieving 358.62 FPS embedding at 720p and 266.67 FPS embedding at 1080p. These results demonstrate that DiM can generalize to practical high-resolution long-video scenarios without additional training.

We also evaluate multi-channel mask encoding under this high-resolution long-video setting, using the same temporal mask recovery protocol as Figure 6. The results follow the same trend as the standard-resolution case: when the mask area exceeds 10%, the multi-channel codes are almost perfectly recovered under all frame-level distortions, indicating that temporal identity recovery remains effective beyond short low-resolution clips.

Table 3: High-resolution long-video evaluation on 720p and 1080p videos with 100+ frames.

Resolution	PSNR	SSIM	Clean	Common Distortions		Video Distortions		Efficiency	
				Valuemetric	Geometric	Frame-level	Compression	Embed FPS	Extract FPS
720p	44.28	0.9928	100	100	96.01	100	100	358.62	170.49
1080p	44.27	0.9932	100	100	94.36	100	100	266.67	167.74

D MASK SHIFTING STRATEGY

Algorithm 1 generates a spatiotemporal mask sequence by propagating a single 2D mask across time using randomized spatial shifts. At each frame, a displacement direction and magnitude are sampled, and the current mask is shifted accordingly. The shifted mask is accepted if it remains non-empty after boundary handling, ensuring valid mask propagation. Repeating this process yields a temporally coherent mask sequence that emulates object-tracking–like behavior while avoiding degenerate empty masks.

Algorithm 1 Spatiotemporal Mask Sequence Generation

Input: Initial 2D mask $\mathbf{M}_0$, Sequence length T, Max movement $\delta_{\max}$
Output: Spatiotemporal mask sequence $\mathcal{S} = \{\mathbf{M}_0, \ldots, \mathbf{M}_{T-1}\}$
Initialize sequence $\mathcal{S} \leftarrow \{\mathbf{M}_0\}$, set current mask $\mathbf{M}_{\text{curr}} \leftarrow \mathbf{M}_0$
for $t = 1$ **to** $T - 1$ **do**
  # Initialize and shuffle candidate directions $\mathcal{D}$
  $\mathcal{D} \leftarrow \{(x, y) \in \{-1, 0, 1\}^2 \mid (x, y) \neq (0, 0)\}$
  Randomly permute the order of $\mathcal{D}$
  for each direction $\mathbf{d} \in \mathcal{D}$ **do**
    # Sample magnitude and compute displacement
    Sample step size $k \sim \mathcal{U}_{\text{int}}(0, \delta_{\max})$
    Compute shift vector $(\Delta x, \Delta y) \leftarrow k \cdot \mathbf{d}$
    # Apply spatial shift and check boundary validity
    $\mathbf{M}' \leftarrow \textsc{Shift}(\mathbf{M}_{\text{curr}}, \Delta x, \Delta y)$
    # Accept if the mask is not empty (contains active pixels)
    if $\sum \mathbf{M}' > 0$ **then**
      $\mathbf{M}_{\text{curr}} \leftarrow \mathbf{M}'$
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{M}_{\text{curr}}\}$
      break
    end if
  end for
end for
return $\mathcal{S}$

Function $\textsc{Shift}(\mathbf{X}, \Delta x, \Delta y)$:
 # Initialize empty tensor
Initialize $\mathbf{X}_{\text{new}} \leftarrow \mathbf{0}$
Let $\Omega = [0, H - 1] \times [0, W - 1]$ be the spatial bounds
 # Get all active indices
Let $\mathcal{I} = \{(c, y, x) \mid \mathbf{X}_{c,y,x} \neq 0\}$ be the set of active indices
 # Parallel coordinate shift
Let $\mathcal{I}' = \{(c, y + \Delta y, x + \Delta x) \mid (c, y, x) \in \mathcal{I}\}$
 # Filter boundary and assign values
Identify valid subset $\mathcal{I}'_{\text{valid}} = \{(c, y', x') \in \mathcal{I}' \mid (y', x') \in \Omega\}$
Map values from $\mathbf{X}$ to $\mathbf{X}_{\text{new}}$ at indices $\mathcal{I}'_{\text{valid}}$
return $\mathbf{X}_{\text{new}}$

Table 4: Instantiation of DiM on the REVMark backbone under different dimensional mappings $\mathrm{REV}\{d_e, d_d\}$. Checkmarks denote qualitative capabilities, where L-E, L-X, and Loc indicate local embedding, local extraction, and tamper localization, respectively.

Method	PSNR	SSIM	Clean	Common Distortions		Video Distortions		Capabilities		
				Valuemetric	Geometric	Frame-level	Compression	L-E	L-X	Loc
$\mathrm{REV}\{1,3\}$	38.27	0.9848	100.00	100.00	93.73	99.77	99.98	✗	✓	✓
$\mathrm{REV}\{2,3\}$	39.36	0.9860	100.00	100.00	95.95	99.61	99.93	✓	✓	✓
$\mathrm{REV}\{3,3\}$	38.95	0.9879	100.00	100.00	97.86	99.95	100.00	✓	✓	✓

E Further Analysis

E.1 Instantiation on the REVMark Backbone

To demonstrate that DiM generalizes beyond the MaskWM backbone, we further instantiate it on REVMark. As shown in Table 4 and Figure 9, the REVMark-based instantiations acquire functionalities beyond conventional copyright protection, including local embedding and tamper localization, which are absent from the original REVMark framework. These results show that such capabilities are induced by the proposed dimensional mappings rather than tied to a specific backbone design. Although their absolute performance is slightly lower than that of the MaskWM-based configurations, this is expected due to differences in backbone capacity. Overall, the results confirm the generality of DiM across different watermarking architectures.

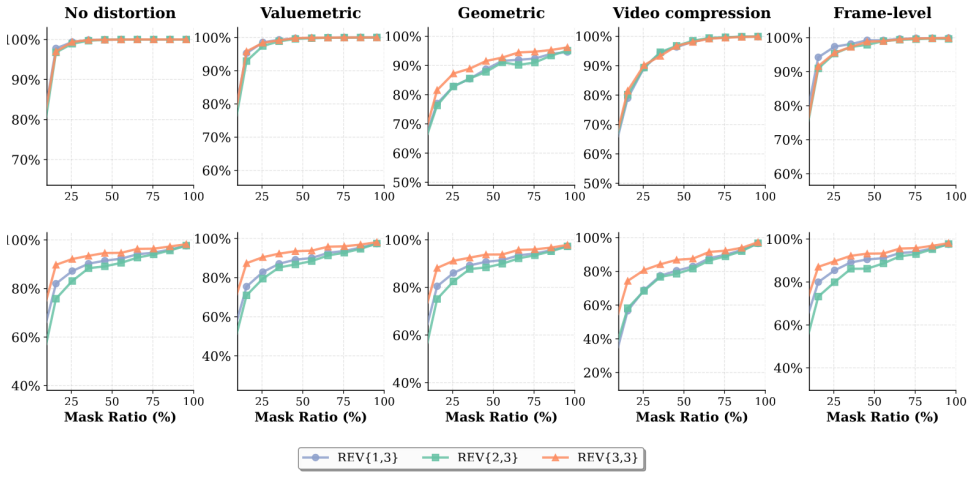

Figure 9: Local extraction (top) and tamper localization (bottom) of REVMark-based dimensional mappings. Results are evaluated under five distortion scenarios, and we select the average value for each interval's ratios to stand for the interval (*e.g.*, 5% represents the average over the 1–10% interval).

E.2 Compression Robustness Enhancement

We observe that increasing payload capacity naturally degrades robustness to compression. To mitigate this effect, we explore a targeted fine-tuning strategy that incorporates Variational Autoencoder (VAE)-simulated distortions into the training process. As shown in Table 5, this approach yields a substantial 6.18% improvement in compression robustness while introducing only minor trade-offs under other distortions.

Table 5: Effect of compression-oriented fine-tuning on robustness and imperceptibility. We report the trade-offs between improved compression robustness and performance under other distortions. Both strategies achieve 100% extraction accuracy under no distortion.

	PSNR ↑	**SSIM ↑**	**No Distortion**	**Common Distortions ↑**		**Video Distortions ↑**	
				Valuemetric	Geometric	Frame-level	Compression
Before	42.36	0.9855	100	100	99.73	100	90.11
After	39.62 (-2.90)	0.9765 (-0.0101)	100 (+0.00)	100 (+0.00)	98.95 (-0.78)	100 (+0.00)	96.29 (+6.18)

E.3 Efficiency

Table 6 compares the efficiency of different methods measured by frames per second (FPS). DiM-V achieves 114.66 FPS for embedding and 228.57 FPS for extraction, surpassing all baseline methods. The substantially higher extraction throughput highlights the suitability of DiM-V for real-time video watermarking scenarios.

Table 6: Comparison of efficiency with baseline methods in terms of frames per second (FPS).

Method	**Image Watermarking**					**Video Watermarking**			
	MaskWM	WAM	OmniGuard	Robust-Wide	TrustMark	VideoSeal	RivaGAN	REVMark	**Ours**
Embed	48.2793	38.4982	3.1480	59.2122	87.5891	87.7029	2.2787	59.7755	**114.6625**
Extract	19.0688	28.0508	0.6189	18.1917	35.3368	97.6562	2.4554	83.7395	**228.5714**

F Ablation and Functional Validation

We summarize the ablation design of DiM from three complementary perspectives. Rather than treating ablation only as the removal of individual modules, our experiments validate whether the proposed dimensional mappings are responsible for the resulting watermarking capabilities.

Backbone-level validation. We first instantiate DiM with different backbone architectures, including REVMark- and MaskWM-based implementations. Across both backbones, changing the dimensional mapping consistently leads to different watermarking capabilities, such as local embedding, local extraction, and tamper localization. This indicates that these behaviors mainly arise from the dimension-aware mapping design rather than a specific encoder–decoder architecture.

Mapping-level ablation. Different choices of $\mathcal{M}\{d_e, d_d\}$ serve as direct ablations of the proposed design space. As summarized in Table 7, these mappings differ in the dimensionalities of the payloads used for embedding and extraction, and consequently induce different functional capabilities. This comparison isolates how embedding–extraction dimensionalities determine the resulting watermarking behavior.

High-to-low mapping ablation. We further ablate $\mathcal{M}\{3, 3\}$ and $\mathcal{M}\{3, 2\}$ to validate the high-to-low mapping. As shown in Figure 5, directly extracting high-dimensional 3D masks under $\mathcal{M}\{3, 3\}$ becomes less robust as the channel dimension increases. In contrast, $\mathcal{M}\{3, 2\}$ decouples spatiotemporal payload extraction into per-frame 2D predictions, substantially improving localization robustness. Together, these ablations validate the role of DiM in systematically guiding watermarking capability design through dimension-aware payload mappings.

Table 7: Mapping-level ablation of DiM. Different embedding–extraction dimensionalities and payload-handling designs induce different watermarking capabilities. L-E, L-X, and Loc. indicate local embedding, local extraction, and tamper localization, respectively.

Mapping	Encoder Input	Decoder Output	L-E	L-X	Loc.
$\mathcal{M}\{1,3\}$	Video + 1D msg	1D msg + 3D mask	✗	✓	✓
$\mathcal{M}\{2,3\}$	Video + 1D msg + 2D mask	1D msg + 3D mask	✓	✓	✓
$\mathcal{M}\{3,3\}$	Video + 1D msg + 3D mask	1D msg + 3D mask	✓	✓	✓
$\mathcal{M}\{3,2\}$	Video + 1D msg + 3D mask	1D msg + 2D masks	✓	✓	✓

G More Visual Results

G.1 Global Watermarking

- $\mathcal{M}\{1,3\}$: see Figure 10.
- $\mathcal{M}\{2,3\}$: see Figure 11.
- $\mathcal{M}\{3,3\}$: see Figure 12.

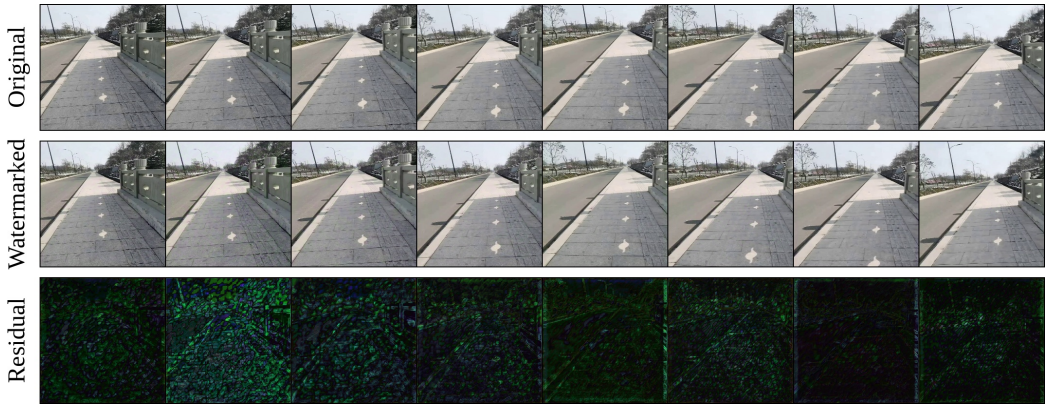

Figure 10: Visualization results of global watermark embedding using $\mathcal{M}\{1,3\}$. The residual images are magnified by $2\times$.

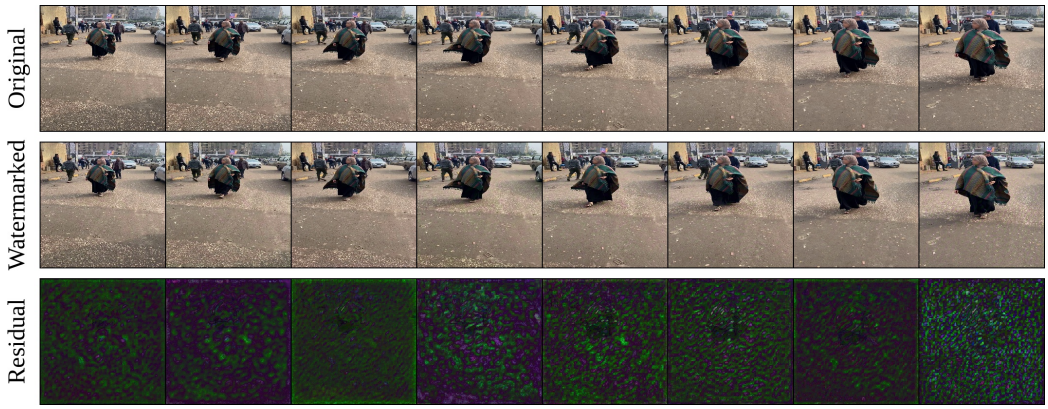

Figure 11: Visualization results of global watermark embedding using $\mathcal{M}\{2,3\}$. The residual images are magnified by $2\times$.

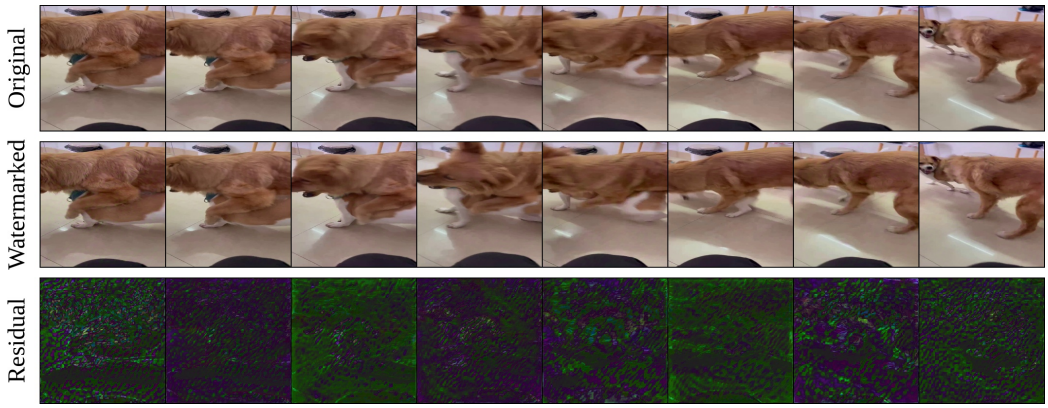

Figure 12: Visualization results of global watermark embedding using $\mathcal{M}\{3, 3\}$. The residual images are magnified by $2\times$.

G.2 Local Watermarking

- $\mathcal{M}\{2, 3\}$: see Figure 13.
- $\mathcal{M}\{3, 3\}$: see Figure 14.

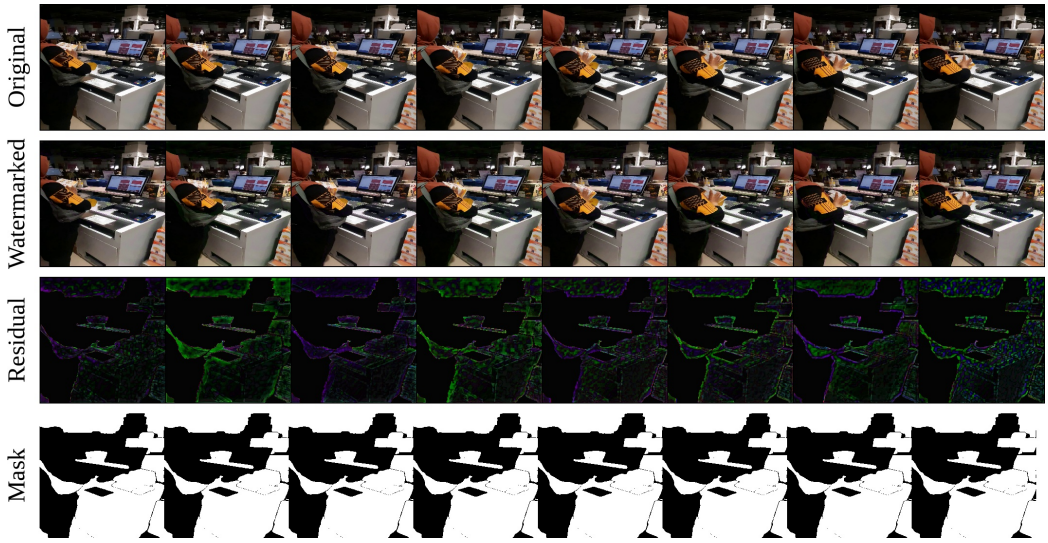

Figure 13: Visualization results of local watermark embedding using $\mathcal{M}\{2, 3\}$. The residual images are magnified by $2\times$.

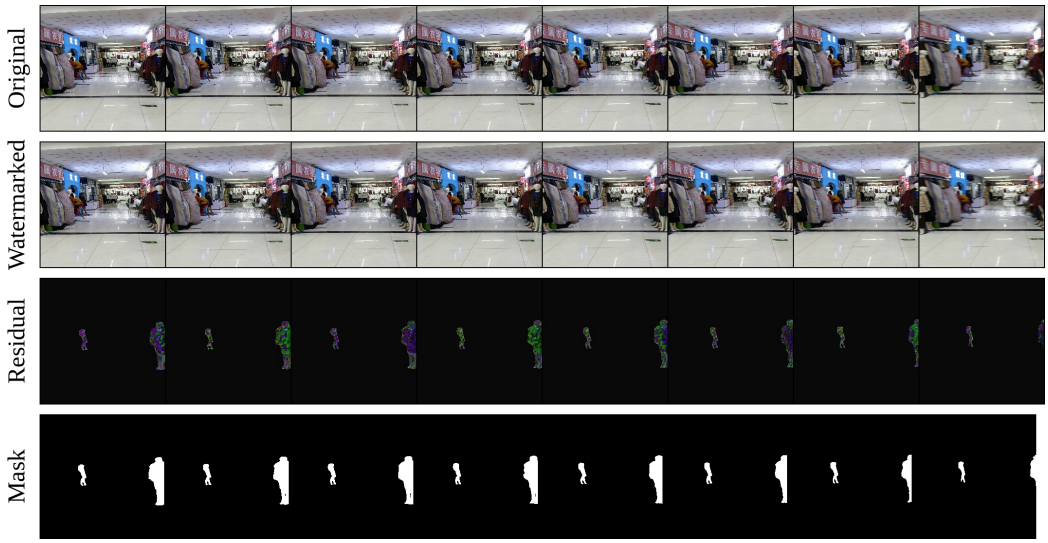

Figure 14: Visualization results of local watermark embedding using $\mathcal{M}\{3, 3\}$. The residual images are magnified by $2\times$.

G.3 LOCALIZATION

- No distortion: see Figure 15.
- Gaussian noise: see Figure 16.
- Salt-and-pepper noise: see Figure 17.
- Median filtering: see Figure 18.
- Gaussian blur: see Figure 19.
- Video compression: see Figure 20.
- Rotation: see Figure 21.
- Perspective transformation: see Figure 22.
- Horizontal flipping: see Figure 23.
- Frame dropping: see Figure 24.
- Frame insertion: see Figure 25.

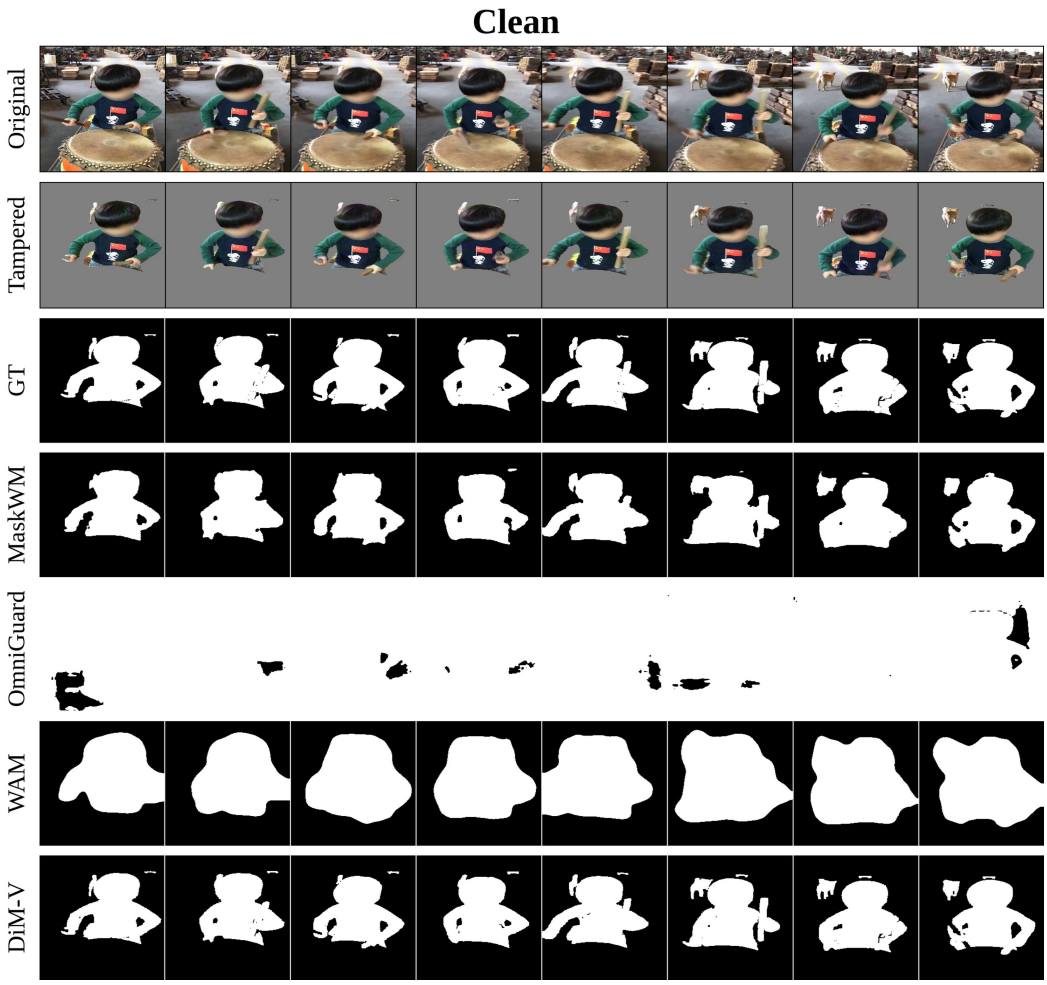

Figure 15: Visualization results of watermark localization using different methods.

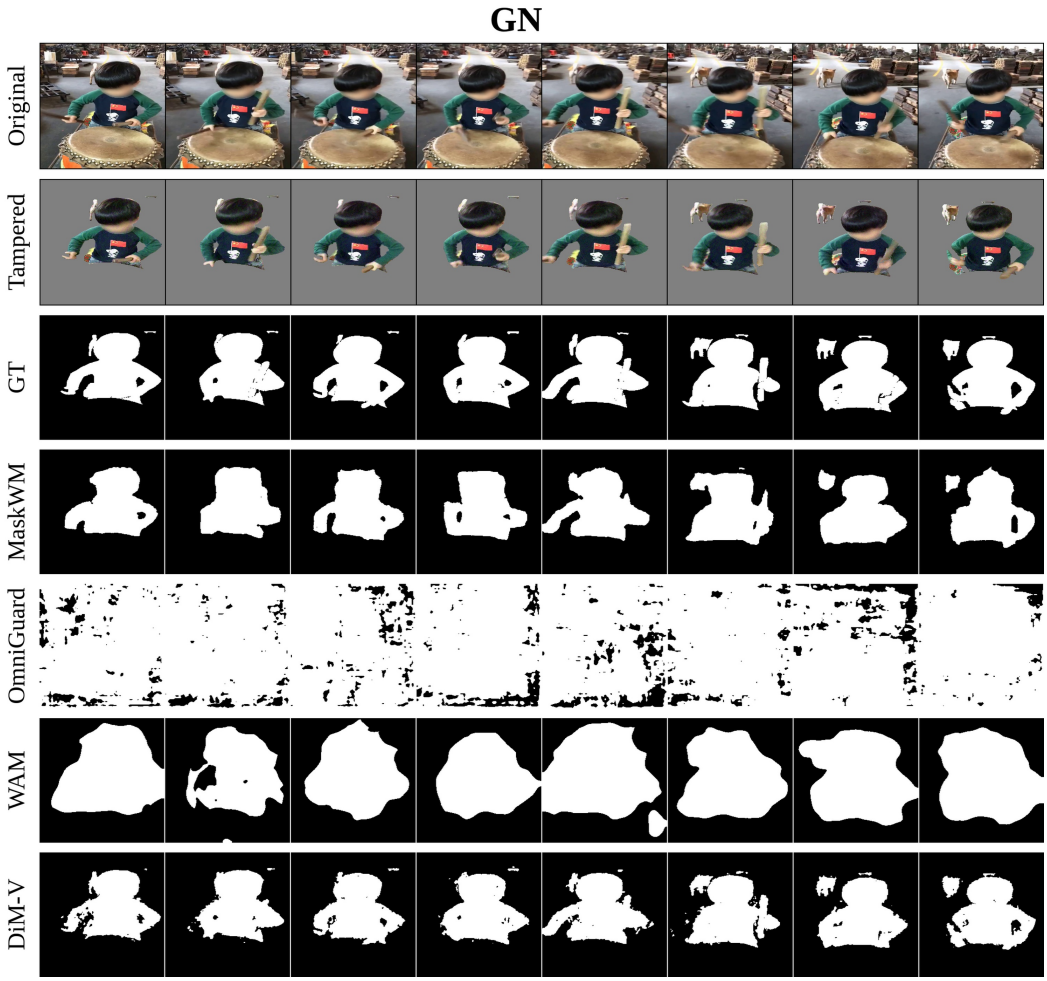

Figure 16: Visualization results of watermark localization using different methods under Gaussian noise.

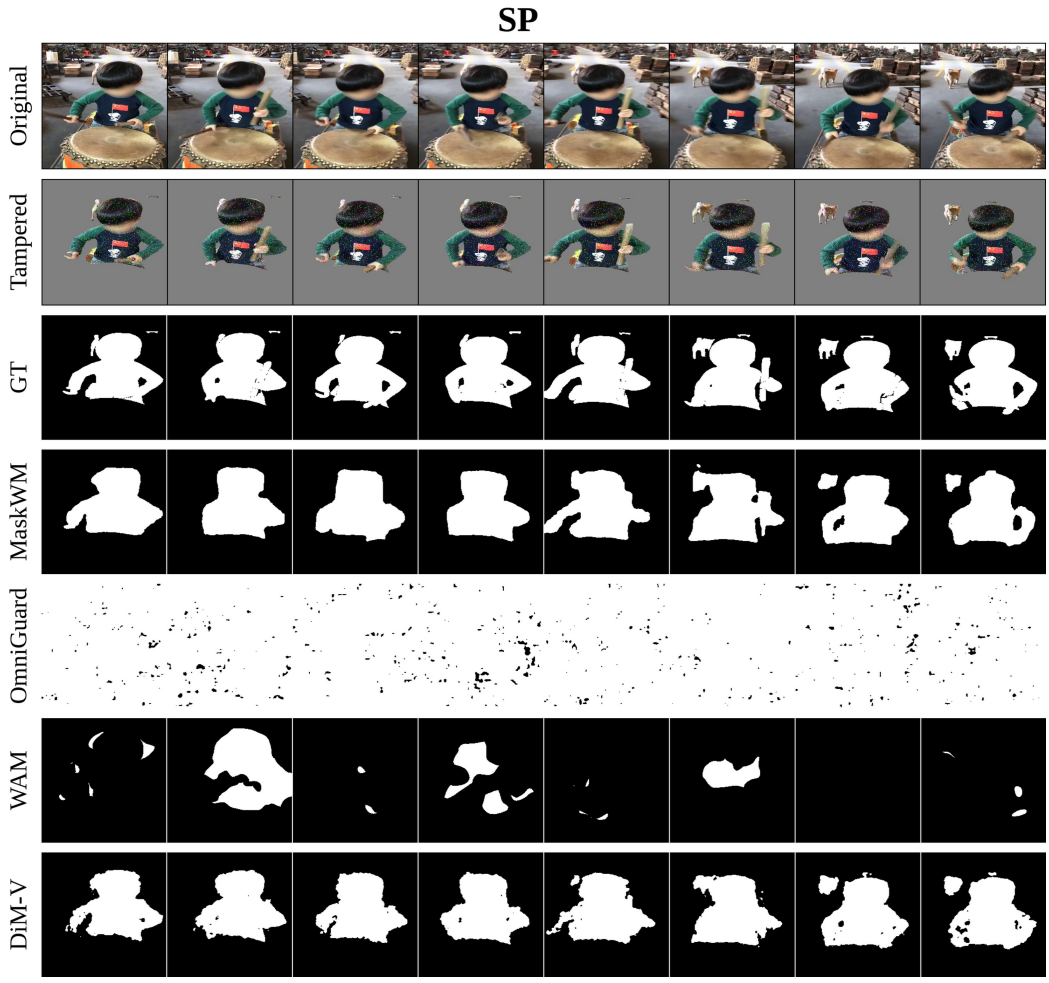

Figure 17: Visualization results of watermark localization using different methods under salt-and-pepper noise.

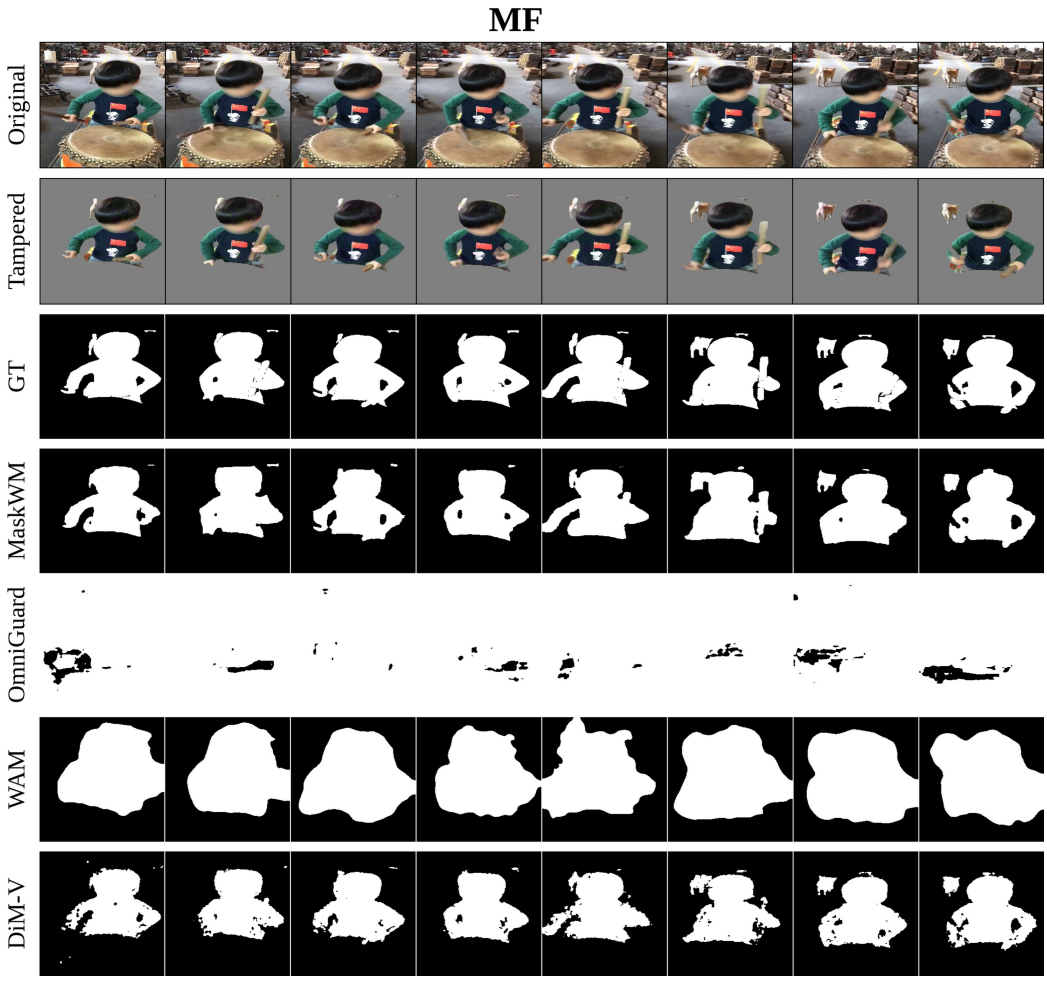

Figure 18: Visualization results of watermark localization using different methods under median filtering.

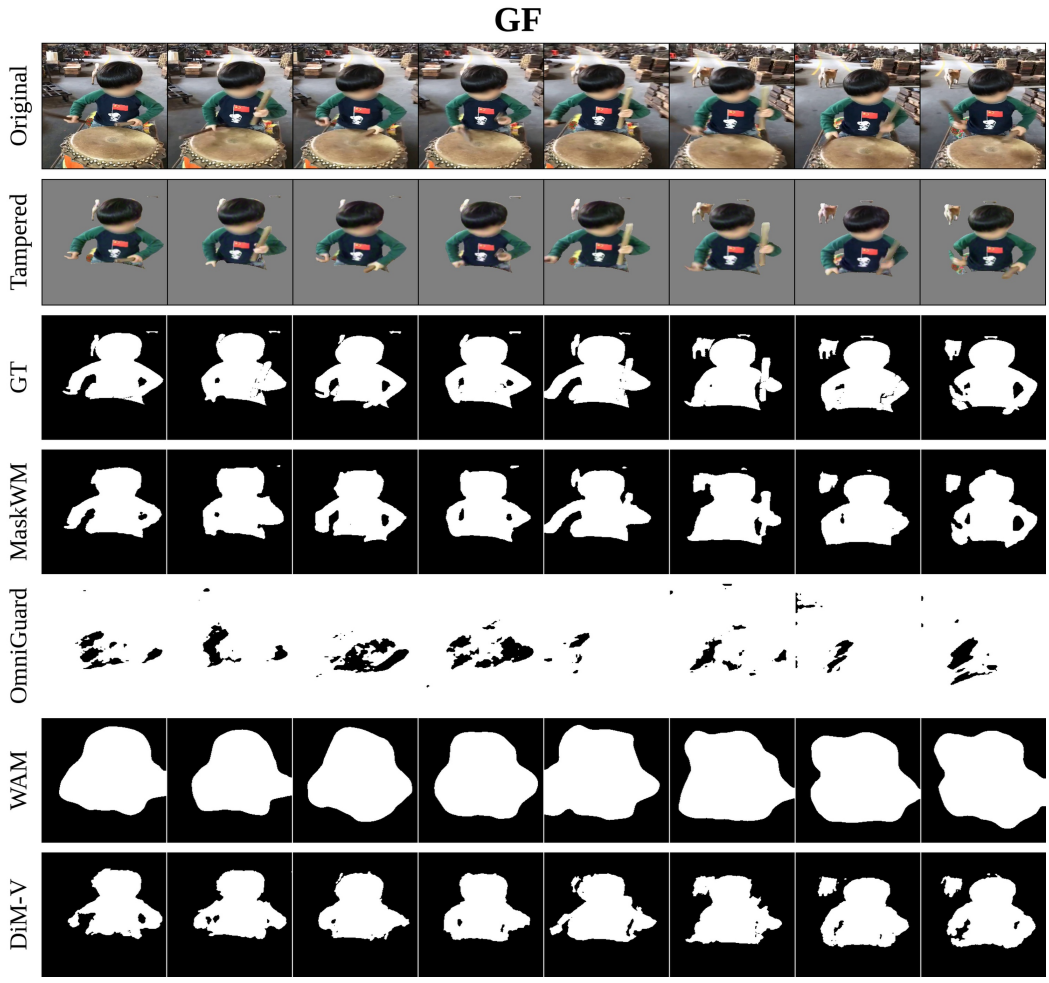

Figure 19: Visualization results of watermark localization using different methods under Gaussian blur.

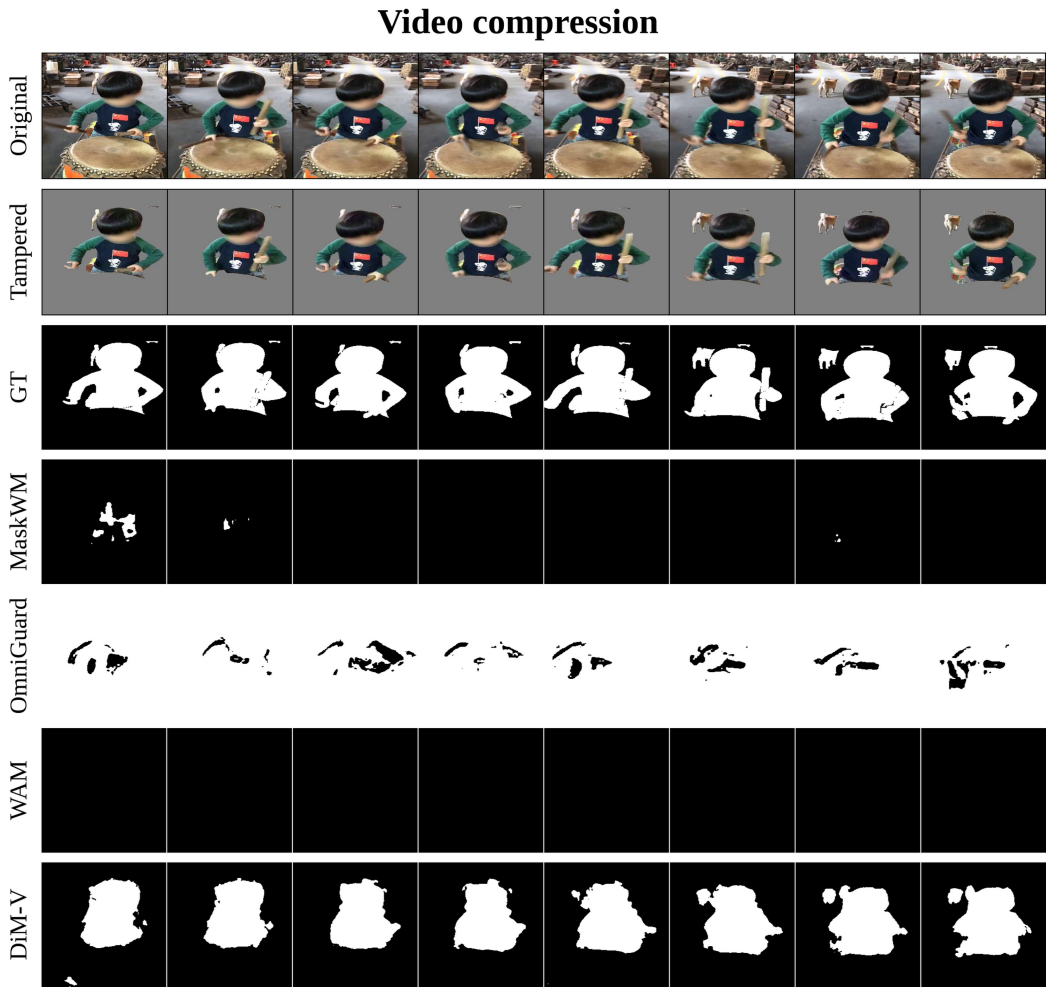

Figure 20: Visualization results of watermark localization using different methods under video compression.

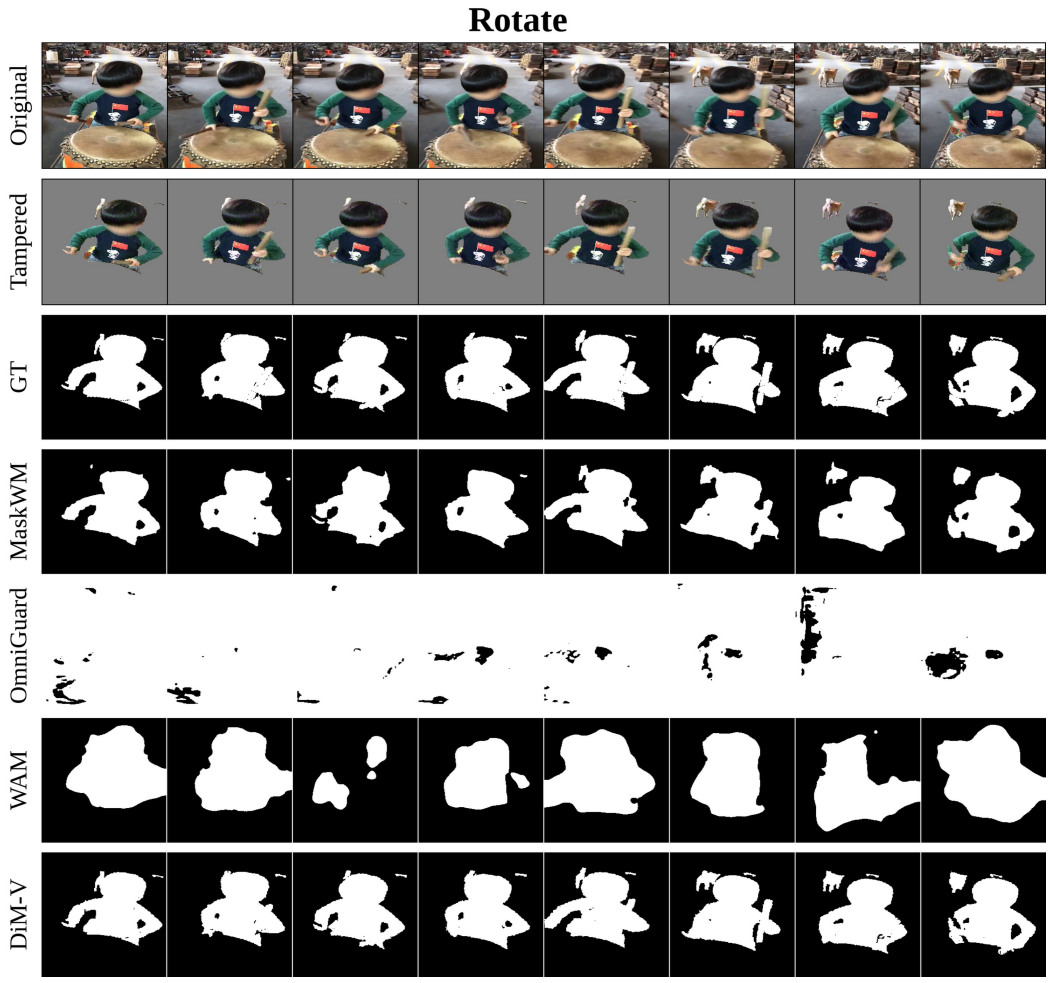

Figure 21: Visualization results of watermark localization using different methods under rotation.

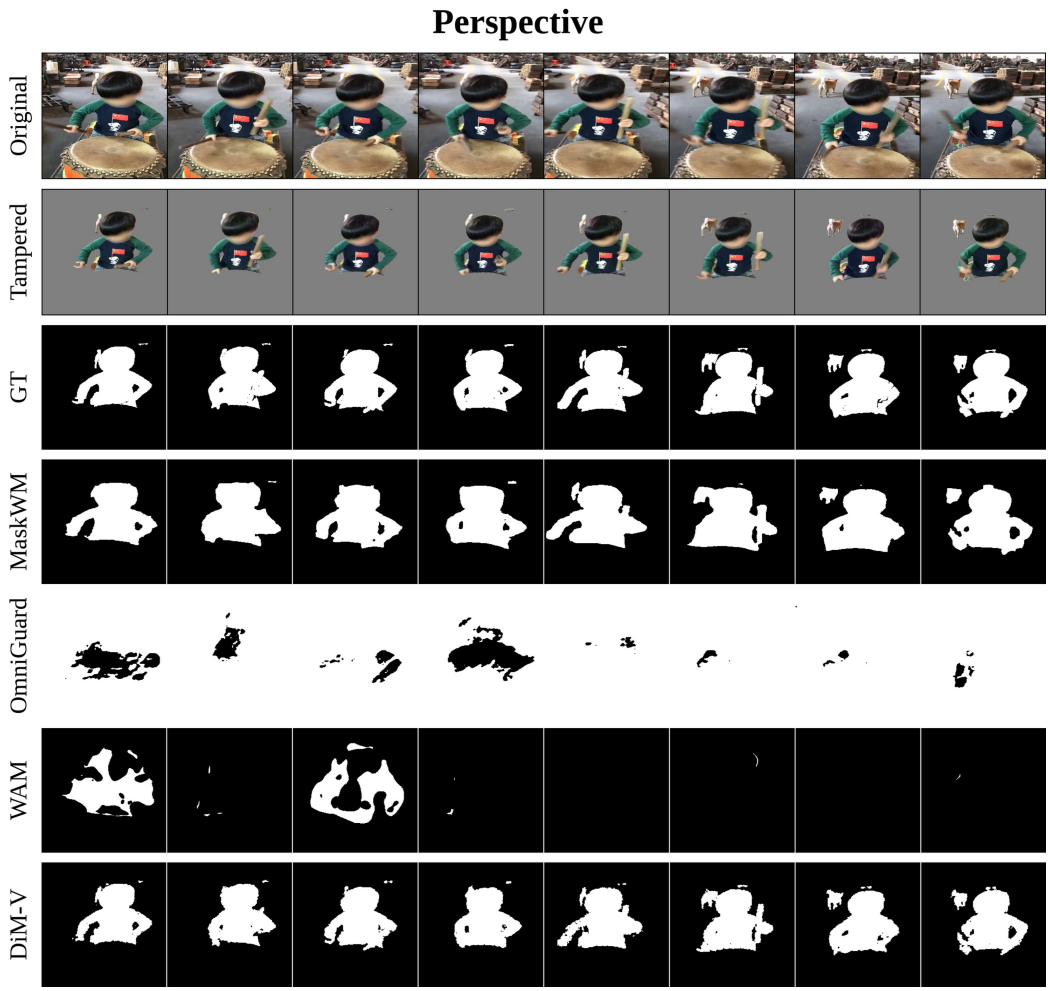

Figure 22: Visualization results of watermark localization using different methods under perspective transformation.

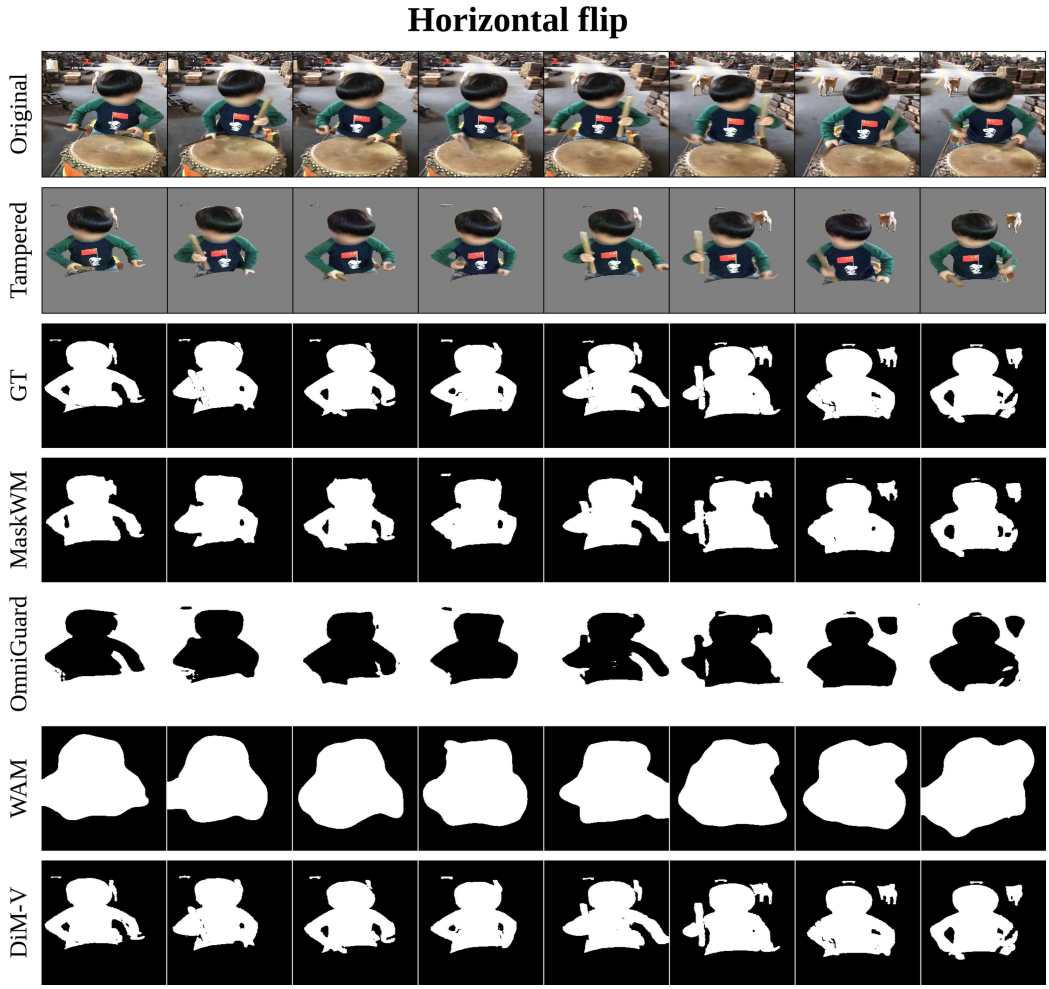

Figure 23: Visualization results of watermark localization using different methods under horizontal flipping.

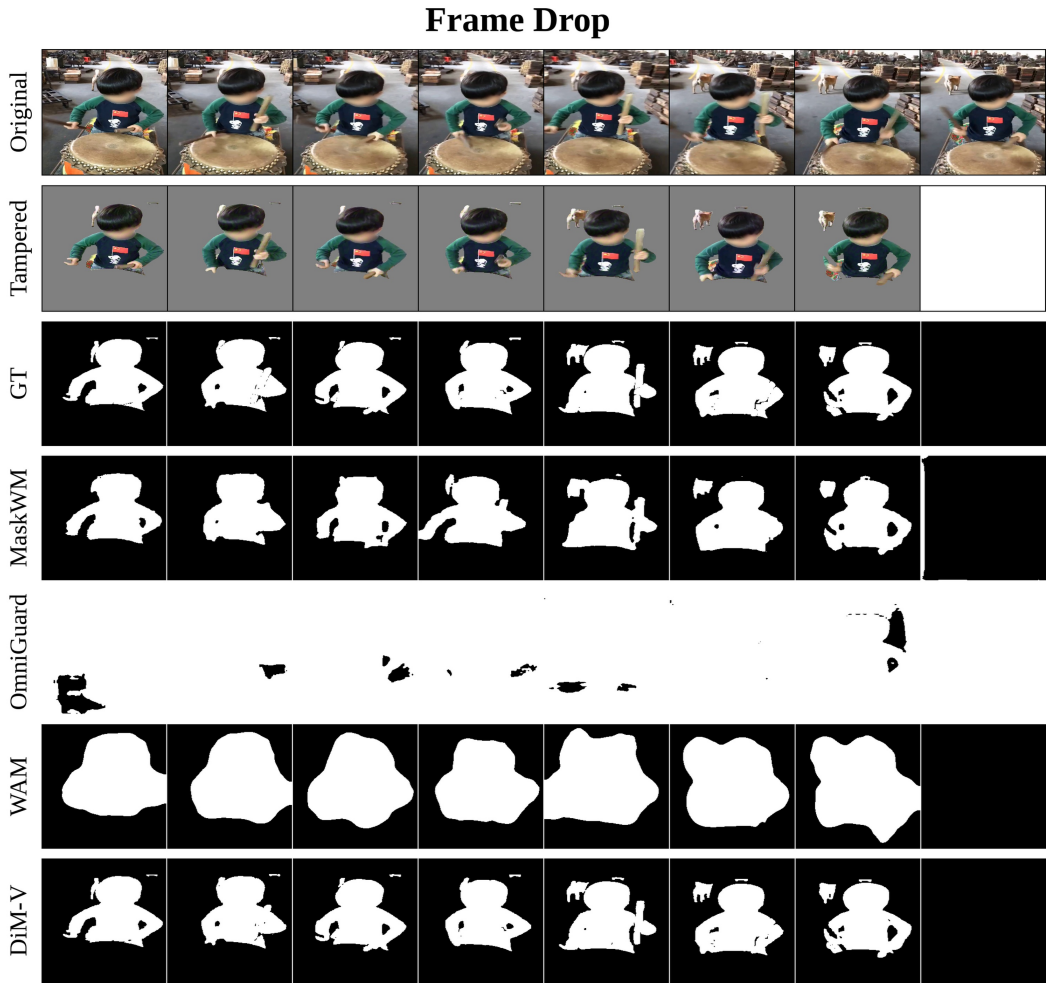

Figure 24: Visualization results of watermark localization using different methods under frame dropping.

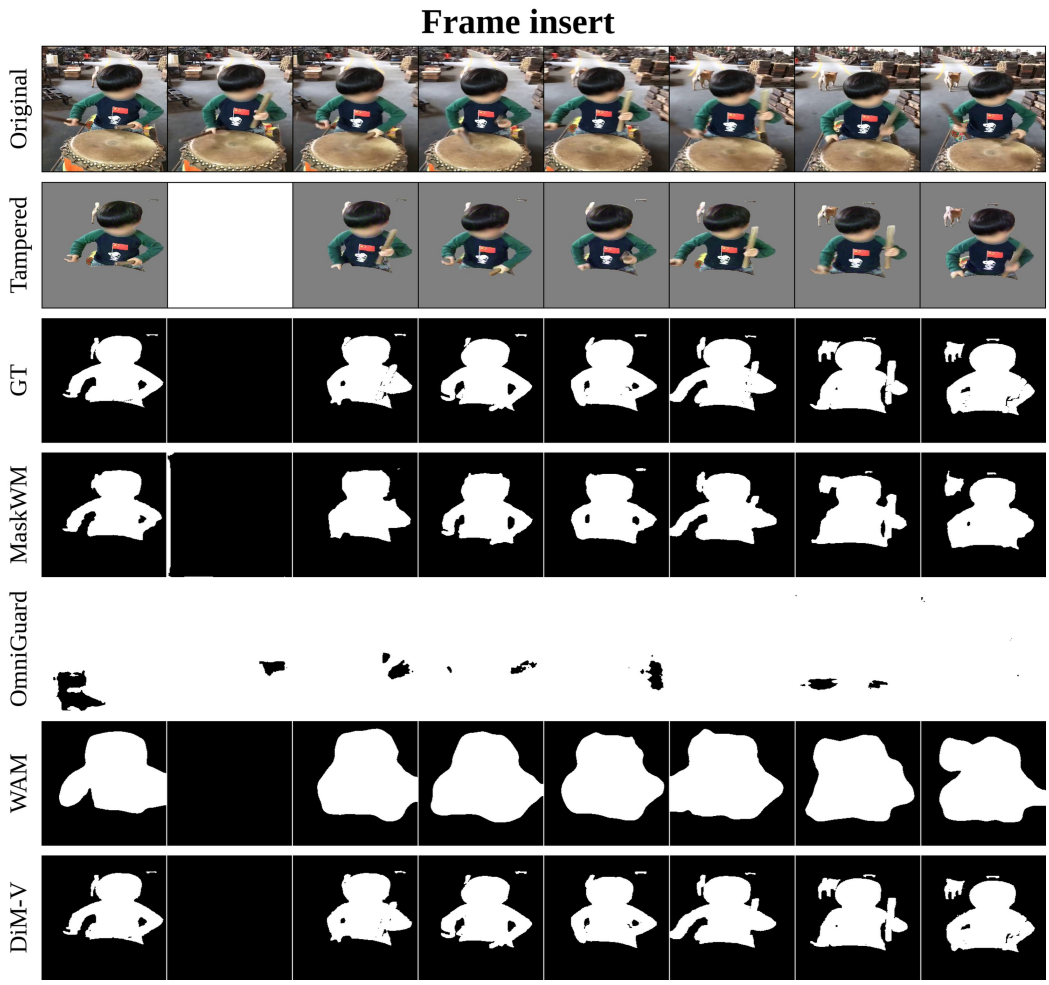

Figure 25: Visualization results of watermark localization using different methods under frame insertion.

G.4 Comparison of $\mathcal{M}\{3,3\}$ and $\mathcal{M}\{3,2\}$ in Multi-Channel Mask Prediction

- No distortion: see Figure 26.
- Under rotation: see Figure 27.

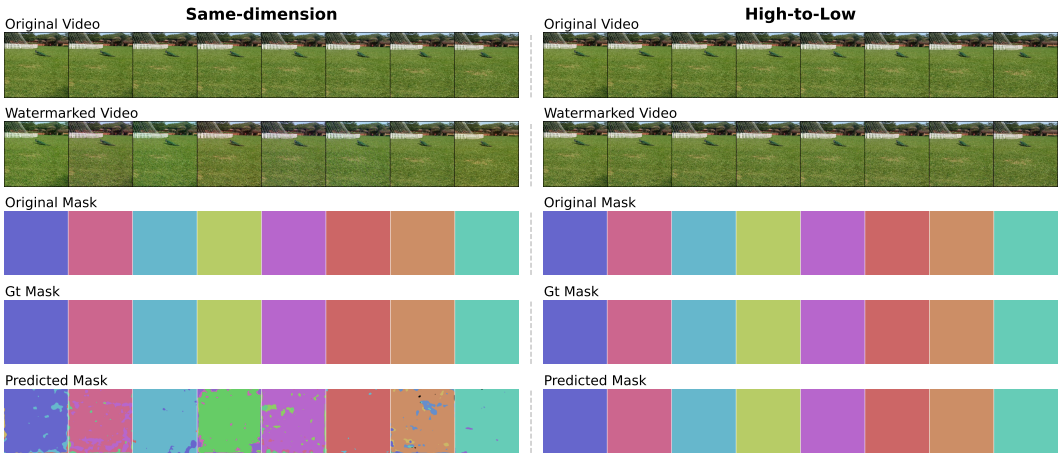

Figure 26: Visualization of predicted multi-channel masks for $\mathcal{M}\{3,3\}$ and $\mathcal{M}\{3,2\}$. Different mask encodings are visualized using distinct colors.

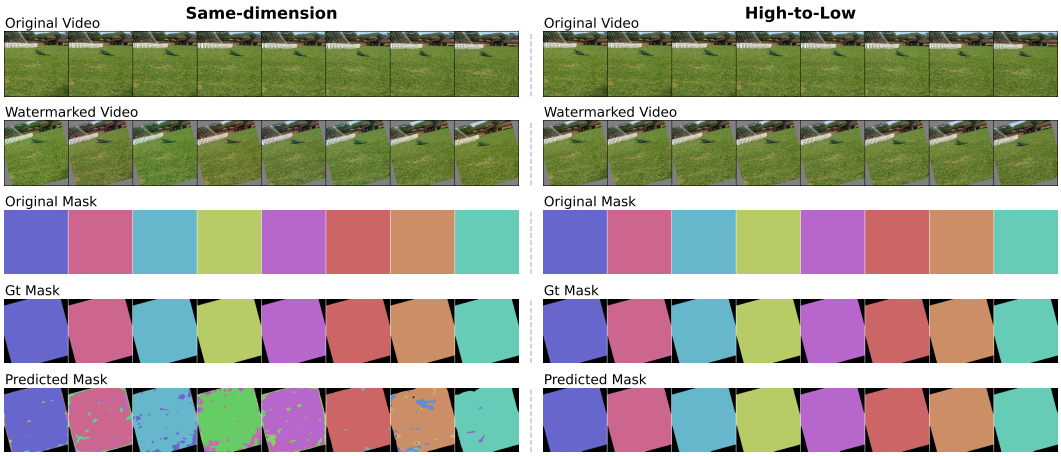

Figure 27: Visualization of predicted multi-channel masks for $\mathcal{M}\{3,3\}$ and $\mathcal{M}\{3,2\}$ under rotation. Different mask encodings are visualized using distinct colors.

H Impact Statement

This work presents a dimension-aware framework for understanding deep watermarking by modeling embedding and extraction as mappings between payloads of different dimensionalities. The framework is analytical and does not introduce new attack capabilities. Instead, it provides a unified perspective for interpreting existing watermarking methods and clarifying how different functionalities arise from representational choices.

When instantiated in video watermarking, the framework shows that capabilities such as spatial and spatiotemporal tamper localization and temporal order recovery can be achieved without architectural changes. These findings are relevant to applications in copyright protection, media authentication, and forensic analysis, and support more principled and interpretable watermarking system design.