Two-Level IVC: Scalable Zero-Knowledge Proof for Trustworthy Video Stream Verification

Fenhua Bai, Xianlin Yang, Tao Shen, *Senior Member, IEEE*, Bei Gong, *Member, IEEE*, Muhammad Waqas, *Senior Member, IEEE*, Hisham Alfuhaid, *Member, IEEE*

***Abstract*—In content-sensitive applications such as video journalism, forensic video authentication, and digital media provenance verification, there is a growing demand for AI analysis of video streams. However, this creates a profound contradiction between data privacy protection and trust in computational processes. Existing solutions struggle to provide scalable cryptographic proofs with temporal integrity for AI computational processes involving complex non-linear operations while simultaneously protecting video content privacy. To address this challenge, this paper proposes an end-to-end Two-Level Incremental Verifiable Computation (IVC) architecture that verifies both intra-frame spatial computation and inter-frame temporal continuity. The architecture's pluggable design instantiates multiple Verifiable Computational Modules (VCM) ranging from pixel-wise transformations and spatiotemporal convolution to Vision Transformer inference, without altering the recursive outer structure. At the hardware deployment level, an adjustable block size mechanism accommodates specific hardware resource constraints by flexibly trading proof generation time against peak memory consumption, achieving constant $O(1)$ peak memory overhead with respect to video length on commodity hardware. Additionally, our framework incorporates the Coalition for Content Provenance and Authenticity (C2PA) standard, ensuring complete sensor-to-proof traceability without requiring hardware trust at the processing level.**

***Index Terms*—Privacy Preserving Video Stream Processing, Zero-Knowledge Proof, Incremental Verifiable Computation, Temporal Integrity, C2PA Content Provenance**

I. INTRODUCTION

With the widespread deployment of deep neural networks in critical video technology domains such as autonomous driving, intelligent surveillance, and medical image analysis, their applications have extended from efficiency enhancement to safety-critical decision support. However, this advancement has brought severe dual challenges of trust and privacy. At the technical level, verifying whether a model deployed on remote or untrusted devices honestly and accurately executes its design specifications is extremely difficult [1], [2]; at the data level, this verification process must protect user privacy from compromise.

For example, in video content, traditional content security technologies such as media forensics and digital watermarking primarily focus on static analysis and integrity verification of media data in controlled environments. However, in the social media era, media objects undergo complex dynamic sharing processes (including platform compression, resizing, re-encoding, and other operations). These dynamic processing procedures both weaken the effectiveness of traditional verification methods and introduce new detectable features, thereby posing new challenges to content security technologies. For instance, when the same image is shared across platforms such as Facebook, Twitter, and WhatsApp, its dimensions, compression quality, and quantization table parameters change significantly [3], rendering traditional PRNU (Photo Response Non-Uniformity) noise detection methods nearly ineffective in such environments.

Digital watermarking can trace video sources but cannot prevent attackers from maliciously executing incorrect analysis algorithms or tampering with computational results on authentic videos [4]. Similarly, deepfake detection technologies are essentially probabilistic detection methods that can determine whether a video is likely forged, but cannot provide mathematically definitive proof that every step of inference in a computational pipeline on video streams strictly follows preset specifications [5], [6]. These methods cannot generally provide formal, cryptographically secure verification of computational processes. Therefore, there is an urgent need for a verification mechanism that provides cryptographic-level security guarantees across the entire computational pipeline of video analysis while protecting privacy.

By leveraging mechanisms such as universal preprocessing and modular constructions (e.g., Polynomial Holographic Proofs), Zero-Knowledge Proofs (ZKP) like zk-SNARKs offer a novel paradigm for verifiable computation, enabling the generation of succinct proofs that can be validated with minimal cost [7], [8]. However, when directly applying the theoretical advantages of ZKP to modern video analysis tasks, feasibility is constrained by two critical technical bottlenecks. First is the computational complexity challenge: the core of modern video analysis lies in deep neural networks, whose powerful learning capabilities stem from nonlinear operations such as ReLU and convolution [9]. Converting these nonlinear operations into ZKP circuits results in a dramatic increase in

Fenhua Bai, Xianlin Yang, and Tao Shen are with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. E-mail: baifenhua@kust.edu.cn; yangxianlin@stu.kust.edu.cn; shentao@kust.edu.cn.

Bei Gong is with the Department of Computing, Beijing University of Technology, Beijing, China. E-mail: gongbei@bjut.edu.cn.

Muhammad Waqas is with the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, United Kingdom. E-mail: engr.waqas2079@gmail.com.

Hisham Alfuhaid is with the Department of Computer Science, King Khalid University, Abha, Saudi Arabia. Email: alasmary@kku.edu.sa.

Tao Shen is the corresponding author.

the number of constraints, leading to enormous performance overhead [10]–[12].

Second is the data scale challenge: videos consist of massive sequences of frames. When employing traditional monolithic SNARK schemes (such as Groth16 [13], PLONK [14]) to verify the processing of a single high-definition video frame, the enormous pixel data volume imposes severe constraints on the complexity of traditional ZKP schemes. Research shows that large circuit tables may contain up to 2^{26} constraint rows [15], while existing methods require very uncommon hardware resources when processing HD resolution images, including 70.7GB memory and 64-core 512GB RAM cloud computing instances [16], making proof generation impractical on commodity hardware.

To address the dual challenges of complexity and scale mentioned above, this paper constructs an end-to-end Two-Level Incremental Verifiable Computation architecture. The design of this architecture is inspired by recursive proof systems [17], [18], achieving cryptographic decoupling of spatial and temporal dimensions by using inner loops to process single-frame chunks and outer loops to link multi-frame commitments.

The core contribution of this article can be summarized in the following three points:

- We propose a Two-Level IVC architecture that simultaneously addresses spatial scalability and temporal integrity in video verification. By decoupling inner and outer recursive loops and committing C2PA content credentials to the IVC initial state, the architecture establishes an end-to-end trust chain from camera-originated hardware attestation to cryptographically verifiable spatiotemporal computation, without relying on trusted execution environments.
- We construct a scalable modular verification paradigm by decoupling the recursive IVC architecture from the core computational logic, instantiated by four Pluggable Verifiable Computational Module (VCM) instances of increasing complexity: grayscale conversion, spatial cropping, spatiotemporal CNN, and Vision Transformer (ViT), demonstrating the framework's generalizability from simple linear operations to complex non-linear models across multiple video processing tasks.
- We introduce an adjustable block size mechanism that enables adaptive deployment on commodity hardware across multiple resolutions. By tuning the block size within the available memory budget, the framework flexibly trades proof generation time against peak memory consumption while preserving $O(1)$ memory complexity with respect to frame count F.

II. RELATED WORKS

Existing trustworthy video processing methods fall into two broad categories: content-centric media security measures, encompassing semantic forensics, passive manipulation detection, and intellectual property protection, and cryptography-based proof systems. This section analyzes the limitations of content-centric security measures in terms of lack of deterministic cryptographic guarantees, as well as the constraints of prior arts on multimedia security and trustworthy verification regarding scalability and verification scope, with particular attention to recent ZKP architectures for image and video authentication and their remaining shortcomings.

A. Limitations of Content-Centric Security Measures

Traditional content security techniques, such as digital watermarking and perceptual hashing [3], [22], primarily target static data integrity. While effective for tracing sources, they fail to guarantee the correctness of complex, dynamic computational processes (e.g., neural network inference); attackers can validly execute malicious algorithms on authentic data without detection. Similarly, deepfake detection methods [23] offer only post-hoc, probabilistic judgments rather than mathematically definitive proof, often lagging behind novel generative models in an adversarial manner [24].

More broadly, these limitations extend from specific video identification methods to a wider range of recent multimedia analytics tasks. For instance, while perceptual video hashing [25] provides robust feature extraction for content identification, its primary shortcoming is the inability to cryptographically verify dynamic computational pipelines on video streams. Similarly, recent application-layer models such as MATCH [26], which performs multi-agent hate video detection, and PNF-Net [27], which localizes image manipulations, operate under the implicit assumption of a trusted execution environment. Their fundamental limitation is the lack of deterministic cryptographic guarantees over the underlying computation. Consequently, when confronted with frame-reordering or host-level data injection attacks, these models can be systematically misled by adversarially constructed media streams, as none of them enforce bit-exact spatiotemporal continuity through verifiable cryptographic mechanisms.

B. Prior Arts on Video Security and Trustworthy Verification

The scalability bottlenecks of monolithic ZKP architectures outlined above have motivated the adoption of Incrementally Verifiable Computation (IVC), which decomposes a long computation into a sequence of short recursive steps and maintains a constant-size accumulator state, thereby decoupling peak memory consumption from total input size. The following representative systems illustrate both the progress and the remaining limitations of IVC-based media verification.

VerITAS [20] applies zero-knowledge proofs to image editing verification, with memory scaling as $O(P)$ for the total image pixels P. Although it eliminates processing-level trust dependencies, it requires up to 127 GB of peak memory for a 30 MP image, which may necessitate cloud-based proof generation in practice, reintroducing a partial dependency on external infrastructure [20]. VIMz [19] advances this by applying IVC to image editing, reducing memory from 300 GB to 3 GB and achieving $O(W)$ memory complexity with $O(\log W)$ proof size through row-wise processing, where W denotes the image width specific to its row-wise scanning framework. However, VIMz relies on only two fixed circuits for row-by-row decomposition, which lacks the generality required for dynamic temporal data. Trust Nobody [16] eliminates

TABLE I
ARCHITECTURAL COMPARISON WITH STATE-OF-THE-ART IMAGE/VIDEO ZKP VERIFICATION SCHEMES

Scheme	Format	Temporal Support	Sensor Trust	Processing Trust	Input Source	Prover RAM	Proof Size	Architecture
VIMz [19]	Image	No	**Required**	No	Raw Pixels	$O(W)$	$O(\log W)$	Fixed
VerITAS [20]	Image	No	**Required**	No†	Raw Pixels	$O(P)$	$O(\log P)$	Fixed
Trust Nobody [16]	Image	No	**Required**	No	Raw Pixels	$O(S_t)$	$O(N_t)$	Fixed
Eva [21]	Video	Full	**Required**	No§	Raw Pixels	$O(1)$	$O(1)$	Fixed¶
Ours	**Video**	**Full**	**Required**	**No**	**Raw Pixels**	$\mathbf{O(1)}$	$\mathbf{O(1)}$	**Flexible**

Sensor Trust (camera hardware attestation) is a universal prerequisite shared by all schemes; Processing Trust is eliminated by construction in all ZKP-based approaches. Input Source: Raw Pixels = direct operation on sensor-originated pixel values. P: total pixel count; S_t/N_t: tile area/count; W: image width.

† VerITAS requires up to 127 GB of peak memory for a 30 MP image, which may necessitate cloud-based proof generation in practice [20], reintroducing a partial dependency on external infrastructure.

§ Eva's ZKP circuit delegates complex encoding steps by ingesting H.264 intermediate data (prediction blocks $\mathbf{P}_{\mathrm{H}}$ and quantized DCT coefficients $\mathbf{Z}_{\mathrm{H}}$) extracted by the verifier, creating a strict functional dependency on the specific H.264 codec architecture.

¶ Eva's architecture is tightly bound to H.264, as its core ZKP circuits are specifically designed for H.264 DCT and quantization structures.

The $O(1)$ prover RAM of our scheme is measured with respect to frame count F. Per-step memory scales as $O(S)$ with block area S.

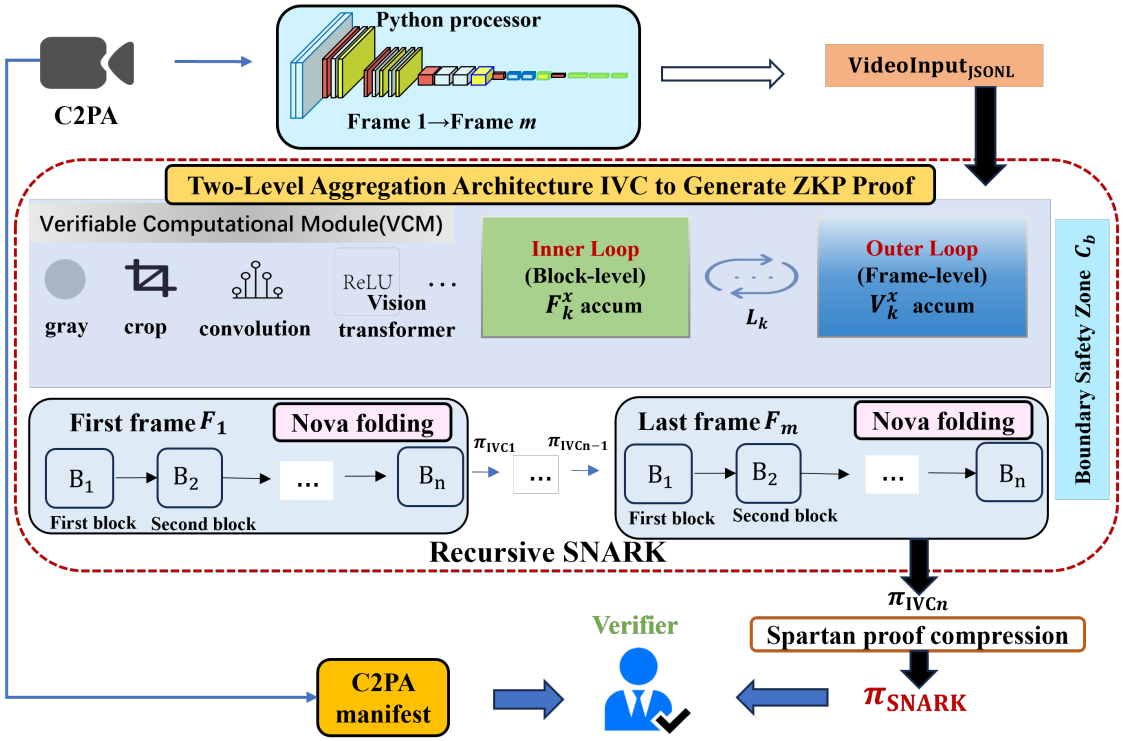

Fig. 1. Overview of the proposed two-level IVC architecture for verifiable video processing. The Boundary Safety Zone corresponds to constraint $\mathcal{C}_b$, which enforces stage-specific bit-width decomposition on all intermediate values to prevent finite-field overflow attacks throughout the computation.

processing-level trust dependencies by decomposing a high-resolution image into non-overlapping tiles and generating an independent zk-SNARK proof per tile, binding each tile to a Poseidon-based Merkle tree root signed by the camera via ECDSA. While this design removes the need for trusted firmware or cloud infrastructure, it operates exclusively on static images without temporal state propagation, and its proof size scales as $O(N_t)$ proportional to the number of tiles N_t, sacrificing succinctness under high-resolution inputs. More critically, VIMz, VerITAS, and Trust Nobody all operate on a per-image basis without temporal state propagation, making them fundamentally incapable of verifying video-specific properties such as inter-frame consistency or motion tracking.

Eva [21] represents the first application of IVC [17] to lossy video authentication, achieving constant $O(1)$ memory and proof size via folding-based IVC, while leveraging intermediate H.264 codec data to reduce circuit complexity. While it mitigates some scaling issues, Eva faces notable limitations that constrain its practical deployment. Specifically, Eva is tightly coupled with H.264 codec structures, confining its applicability to H.264-encoded content and requiring full redesign of core ZKP circuits to support any alternative

codec. Furthermore, Eva's verification correctness relies on the verifier independently re-extracting H.264 intermediate data (prediction blocks and quantized DCT coefficients) from the published video bitstream, creating a hard dependency on H.264 decoder availability at the verification side and limiting applicability to non-H.264 workflows. Moreover, although Eva achieves $O(1)$ prover RAM asymptotically, it still requires up to 60 GB of peak memory in practice, posing significant challenges for deployment on resource-constrained devices [28].

Despite recent progress, as detailed in Table I, existing schemes exhibit complementary but unresolved limitations. Image-based approaches such as VIMz, VerITAS, and Trust Nobody lack temporal state propagation, rendering them incapable of verifying video-specific properties such as inter-frame consistency and frame ordering. Among video-oriented schemes, Eva achieves constant memory by exploiting H.264 codec structures, but this tight coupling restricts its applicability to the H.264 editing pipeline and reintroduces privacy risks by requiring access to intermediate codec data. These limitations collectively highlight the need for a verification framework that supports temporal integrity, maintains constant memory complexity with respect to video length, and operates independently of specific codec structures or hardware dependencies.

III. A Hierarchical Folding Architecture for Verifiable Video Processing

To address the scalability and computational complexity challenges in large-scale video verification, we design an end-to-end Two-Level IVC architecture, as shown in Fig. 1. The architecture begins with videos processed by a Python preprocessor to generate a `VideoInput_JSONL` stream α, in which the first line encodes configuration metadata including the C2PA trust anchor τ_{C2PA}, and subsequent lines encode the per-block pixel stream. This stream is then subjected to frame extraction and block decomposition via pluggable Verifiable Computational Modules (VCMs), instantiated as grayscale conversion, spatial cropping, spatiotemporal CNN, and Vision Transformer in this work.

In the core verification phase, the system leverages recursive proof composition via efficient Nova folding to incrementally generate the proof π_{IVC} through recursive steps, adopting a Two-Level IVC architecture for hierarchical processing where the inner loop handles fine-grained verification of block-level data via the frame accumulator F_k^x, while the outer loop implements cross-frame state aggregation via the video accumulator V_k^x ($x \in \{\text{orig}, \text{mod}\}$). The C2PA trust anchor τ_{C2PA}, committed at state slot $z_0[10]$, binds the genesis block hash to the camera-attested hardware credential, establishing an unbroken trust chain from the physical sensor to the final proof. Finally, as depicted in the bottom portion of Fig. 1, the system compresses the recursive proof into a constant-sized π_{SNARK} through the Spartan algorithm. The Boundary Safety Zone, instantiated as constraint $\mathcal{C}_b$, enforces stage-specific w_k-bit decomposition on all intermediate values and hard-constrains the loop control signal $L_k \equiv (b_k = B - 1)$, providing multi-layered security guarantees throughout the entire process. The verifier authenticates the computation by checking π_{SNARK} against the public states z_0 and z_n, and independently verifying the C2PA [29] manifest extracted from the source video, without requiring access to the raw pixel content, ensuring verifiable computational integrity while protecting video content privacy.

A. Recursive Proof Composition via Efficient Folding

Building on the Nova folding scheme [17] and Pedersen commitments [30], [31] over the Pasta elliptic curve cycle [32], we design a recursive proof composition architecture that efficiently aggregates proofs across arbitrarily long video sequences while maintaining constant verification complexity. This approach eliminates pairing-based cryptography [13] and trusted setup procedures while providing the scalability necessary for practical video verification.

Our architecture leverages the committed relaxed Rank-1 Constraint System (R1CS) [17] framework, which extends traditional constraint systems through controlled relaxation:

$$\mathbf{Az} \circ \mathbf{Dz} = u \cdot \mathbf{Qz} + \mathbf{E} \tag{1}$$

where $u \in \mathbb{F}_p$ is a scalar relaxation parameter and $\mathbf{E} \in \mathbb{F}_p^m$ denotes an error vector that accumulates across folding operations. Unlike traditional strict R1CS systems that enforce exact equality $\mathbf{Az} \circ \mathbf{Dz} = \mathbf{Qz}$, this relaxation enables efficient instance combination while maintaining soundness [17], [33]. Security relies on Pedersen commitments to witness and error vectors over elliptic curve groups, whose homomorphic property enables combining committed values without revealing underlying witnesses while maintaining binding under the discrete logarithm assumption [34]. The core folding operation merges two relaxed R1CS instances through a random linear combination. Given two instances with witness vectors $\mathbf{z}_1, \mathbf{z}_2$ and commitments (W_1, E_1) and (W_2, E_2), the folded witness commitment is:

$$W = W_1 + r \cdot W_2 = \text{Com}(\mathbf{z}_1 + r \cdot \mathbf{z}_2, r_{w_1} + r \cdot r_{w_2}) \tag{2}$$

where the folding challenge $r \in \mathbb{F}_p$ is generated via Poseidon hash [35] using the Fiat-Shamir transform [36]. The error commitment incorporates cross-terms from the bilinear structure:

$$E = E_1 + r \cdot E_2 + r(1 - r) \cdot T \tag{3}$$

where T captures the interaction between folded instances, ensuring the folded instance correctly represents the combined computation [17].

Our Two-Level IVC architecture maps naturally onto this folding scheme structure, recognizing that video processing involves nested computational patterns [37]. The inner layer processes individual video blocks within frames, where each verification step corresponds to a single R1CS instance representing Circom constraints [38] that encode the target computational operation defined by the active VCM module. The witness vector $\mathbf{z}$ for each step encompasses pixel values, intermediate computation results from convolution and activation layers, and the temporal state buffer T_k. For stateful modules such as the spatiotemporal CNN and Vision Transformer, T_k

TABLE II
KEY SYMBOLS AND NOTATIONS

Symbol	Type	Description
Unified IVC State Vector ($d = 11$)		
z_k	$\mathbb{Z}_p^{11}$	Unified 11-dim. state at step k: $(f_k, b_k, V_k^{\text{orig}}, F_k^{\text{orig}}, V_k^{\text{mod}}, F_k^{\text{mod}}, B, p_k, M_k, t_k, \tau_{\text{C2PA}})$.
$f_k,\ b_k$	$\mathbb{Z}_p$	Frame and block indices at $z[0]$, $z[1]$.
V_k^{orig}, V_k^{mod}	$\mathbb{Z}_p$	Dual outer-loop video accumulators at $z[2]$/$z[4]$: V_k^{orig} chains all original frame commitments; V_k^{mod} chains the modified stream (grayscale/crop/CNN/ViT), where $x \in \{\text{orig}, \text{mod}\}$.
F_k^{orig}, F_k^{mod}	$\mathbb{Z}_p$	Dual inner-loop frame accumulators at $z[3]$/$z[5]$: updated each block via Poseidon hashing of block feature hash, block index, and timestamp; reset to 0 at frame boundary, where $x \in \{\text{orig}, \text{mod}\}$.
T_k; h_m	$\mathbb{Z}_p^t$; $\mathbb{Z}_p$	T_k: temporal feature buffer (e.g., H_{t-1} for CNNs); unused in stateless modules. $h_m = \text{Poseidon}(\Theta)$: model-hash constant enforced at $z[4]$ in the CNN/Transformer module only.
L_k; B	$\{0,1\}$; $\mathbb{Z}$	L_k: inner/outer loop flag; hard-constrained as $L_k \equiv (b_k = B-1)$ (not a free prover input). B: verifier-fixed block count per frame at $z[6]$; prevents cross-resolution forgery.
w_k; δ_{start}	$\mathbb{Z}$; $\{0,1\}$	w_k: stage-specific certified bit-width for constraint $\mathcal{C}_b^{(k)}$ (8-bit pixels, 11-bit spatial outputs, 12-bit temporal outputs, 16-bit ViT intermediate values). $\delta_{\text{start}} = \mathbf{1}[f_k{=}0 \wedge b_k{=}0]$: genesis-block indicator used to enforce C2PA anchoring and zero-initialize H_{t-1}.
RLC, C2PA & HR-PRS Extensions		
h_k^{rlc}; ρ_k; η_k	$\mathbb{Z}_p$	η_k: per-step private nonce. $\rho_k = \text{Poseidon}(\eta_k, t_k, F_k^{\text{orig}})$: Fiat-Shamir Random Linear Combination (RLC) challenge. $h_k^{\text{rlc}} = \sum_{i=0}^{P-1} \text{pack}(r_i, g_i, b_i) \cdot \rho_k^i$: RLC commitment for pixel-level modules, where $\text{pack}(r,g,b) = r + g{\cdot}2^{10} + b{\cdot}2^{20}$, $P = H{\times}W$.
τ_{C2PA}	$\mathbb{F}_p$	Camera-attested C2PA credential hash, fixed at $z[10]$ for all steps. Circuit enforces $\delta_{\text{start}} \cdot (h_1^{\text{rlc}} - \tau_{\text{C2PA}}) = 0$, binding the genesis block to the physical sensor.
ρ_{mask}; p_k; M_k/l	$\mathbb{F}_p$	Crop-module spatial mask variables: $\rho_{\text{mask}} = \text{Poseidon}(\tau_{\text{C2PA}}, 1, 0)$; $p_k = \rho_{\text{mask}}^k$ (running power at $z[7]$); M_k (running sum at $z[8]$, boundary enforces $M_{\text{final}} = M_{\text{target}}$); $z[8]$ is repurposed as ViT layer index $l \in \{0, \ldots, N_{\text{layers}} - 1\}$, reset to 0 upon block completion.
n; γ	$\mathbb{Z}$; $(0,1]$	HR-PRS parameters: n = number of pseudo-random probes; γ = adversarial modification coverage ratio. The per-block evasion probability satisfies $P_{\text{evade}} \leq (1-\gamma)^n$.
H_{t-1}^{eff}; H_{prover}	$\mathbb{F}_p^{\lvert H\rvert}$	Effective temporal buffer and prover's unconstrained history witness; zero-initialized at genesis via δ_{start} as formalized in Equation (11).
D_f	$\mathbb{Z}$	Feature vector dimension used in the HR-PRS sampling mechanism, where ϕ_j denotes the j-th element of the feature vector and $j \in \{0, \ldots, D_f - 1\}$.
Folding Scheme & Security		
α	Stream	VideoInput_JSONL stream produced by the Python preprocessor, carrying per-block pixel data and the C2PA trust anchor τ_{C2PA}.
u; E	$\mathbb{F}_p$; $\mathbb{F}_p^m$	Relaxation scalar and error vector in the committed relaxed R1CS: $\mathbf{A}z \circ \mathbf{D}z = u\mathbf{Q}z + E$; m denotes the number of constraints.
z_k^{in}	$\mathbb{Z}_p^{11}$	State vector of the incoming frame at folding step k, as used in $z^{(k+1)} = z^{(k)} + r_k \cdot z_k^{\text{in}}$.
π_{IVC}; π_{IVC_n}	String	Intermediate and final recursive IVC proof accumulating across folding steps (n denotes the total number of IVC steps), prior to Spartan compression into π_{SNARK}.
λ	$\mathbb{Z}$	Cryptographic security parameter; $\text{negl}(\lambda)$ denotes a negligible function in λ.
vk; x	String; $\mathbb{F}_p^*$	Verification key and public input used in the verifier equation $\mathcal{V}(\text{vk}, x, \pi_{\text{SNARK}}) = 1$.
$\mathcal{F}_T$	Function	Video step transformer: $z_{k+1} \leftarrow \mathcal{F}_T(z_k, \Theta)$. Algorithms 1–3 are concrete circuit-level instantiations of $\mathcal{F}_T$, each executed once per video block at every IVC step.
Constraints & Parameters		
$\mathcal{C}_t$, $\mathcal{C}_b$, $\mathcal{C}_{\text{state}}$	Predicate	Temporal continuity ($t_k > t_{k-1}$), boundary safety (w_k-bit decomposition of intermediate values), and state consistency ($z[4] = h_m$).
Θ; t_k	Set; $\mathbb{Z}_p$	Θ: private parameter set (CNN weights Θ_s, Θ_t, crop windows). t_k: strictly monotonic timestamp at $z[9]$; input to Fiat-Shamir challenge ρ_k.
$(\mathbf{A}, \mathbf{D}, \mathbf{Q})$; π	$\mathbb{F}_p^{m \times n}$; String	R1CS constraint matrices encoding $\mathbf{A}z \circ \mathbf{D}z = u\mathbf{Q}z + E$; π: final Spartan-compressed ZK proof.
$\mathcal{R}$	Predicate	R1CS circuit satisfiability predicate; $\mathcal{R}(w') = 1$ denotes that witness w' satisfies all R1CS circuit constraints.
$\mathcal{E}_{\text{forge}}$	Event	Adversarial forgery event; $\Pr[\mathcal{E}_{\text{forge}}] \leq \text{negl}(\lambda)$ captures the security condition against temporal ordering violation, parameter tampering, and state replay attacks.

carries historical features from previous frames, creating cryptographic linkages across frames that are absent in per-frame verification schemes. For stateless modules such as grayscale conversion and cropping, T_k carries no semantic content and is initialised to zero at the genesis block via the δ_{start} constraint. The module-specific update function $\mathcal{U}_{\text{in}}$ is instantiated as the identity map, preserving $T_k = \mathbf{0}$ throughout the proof chain. This design maintains the fixed 11-dimensional state vector required by Nova's folding scheme without introducing additional constraint overhead for stateless operations. The outer layer aggregates frame-level results across the video sequence through iterative folding [17], [39], combining the previously accumulated computation state $(\mathbf{z}^{(k)}, u^{(k)}, \mathbf{E}^{(k)})$ with a new computation step to generate $(\mathbf{z}^{(k+1)}, u^{(k+1)}, \mathbf{E}^{(k+1)})$. Each folding operation effectively compresses the proof for frame k into the accumulated state while adding frame $k{+}1$, yielding the mathematical progression:

$$z^{(k+1)} = 0 + r_k \cdot z_k^{\text{in}} \tag{4}$$

where r_k is a fresh random challenge for each frame and z_k^{in} denotes the state vector of the incoming frame at step k. This recursive aggregation maintains constant proof size regardless of video length (as shown in the Table I), thereby yielding $O(1)$ total prover memory with respect to video

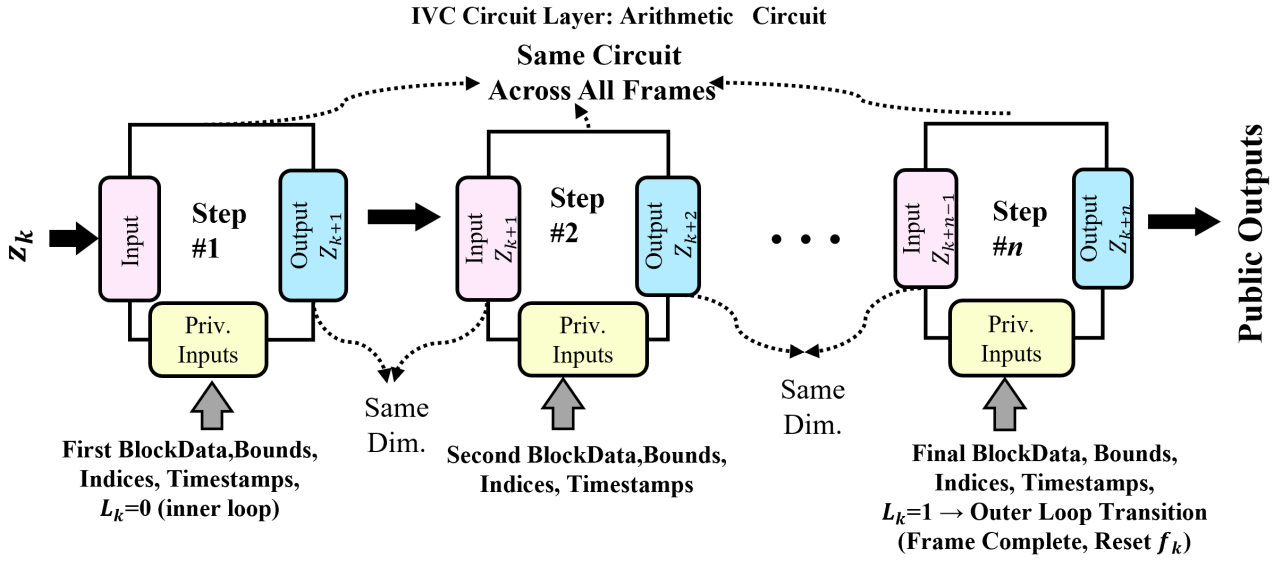

Fig. 2. Recursive folding architecture for Two-Level IVC video verification. The same circuit is applied across all video frames to process blocks incrementally. Each step takes a public state vector $z_k = (f_k, b_k, V_k^{\text{orig}}, F_k^{\text{orig}}, V_k^{\text{mod}}, F_k^{\text{mod}}, B, p_k, M_k, t_k, \tau_{\text{C2PA}})_k$ as input and produces an updated state z_{k+1}, while private inputs contain the actual block data ($H \times W$ pixels), boundaries, indices, and timestamps. The control signal L_k distinguishes between inner loop processing ($L_k = 0$, intra-frame blocks) and outer loop transitions ($L_k = 1$, frame completion with f_k reset).

length F. Critically, the hierarchical structure enables the verification of video-specific properties such as monotonic timestamp progression and module-defined feature consistency across a configurable sliding window, which cannot be expressed in systems treating frames independently. Finally, to achieve ultimate succinctness, our architecture employs a proof-compression step after iterative folding is complete, as depicted in Fig. 1. We use a Spartan-based Compressed-SNARK [19] to compress the verification of the entire F steps (video length) folding process into a single proof ranging from approximately 30 KB depending on the active VCM module, whose verification cost remains constant with respect to video duration regardless of the video's complexity. This final compression transforms the accumulated folding state into a succinct argument that cryptographically commits to the correctness of all F frames while maintaining the zero-knowledge property.

Building on the efficient folding mechanism, we design a Two-Level IVC aggregation architecture to organize the recursive verification process across video frames systematically. The architectural challenge arises from the fundamental mismatch between video structure and cryptographic recursion: videos naturally decompose into spatial blocks within frames and temporal sequences across frames, while traditional flat recursive schemes [12], [40], [41] treat all computational steps uniformly. This mismatch manifests in memory inefficiency, as flat schemes require $O(B{\cdot}F)$ memory to maintain states for all B blocks across F frames, where B represents the number of blocks per frame. Purely spatial operations such as grayscale conversion and cropping exhibit locality within frames, while spatiotemporal operations such as spatiotemporal convolution and Vision Transformer inference require both intra-frame spatial aggregation and cross-frame temporal dependencies. Our Two-Level IVC architecture resolves both issues by decomposing verification into two nested recursive layers, an inner recursion dedicated to intra-frame spatial aggregation and an outer recursion for inter-frame temporal accumulation. This design effectively leverages the Pasta elliptic curve cycle and Nova's committed relaxed R1CS framework [17] to achieve $O(1)$ memory complexity with respect to F while preserving temporal dependencies.

The architectural foundation rests on a carefully designed unified state vector that encapsulates all verification context within a single cryptographic commitment. As depicted in Fig. 2, the architecture operates through recursive application of a single circuit $\mathcal{F}$ to each video block, with the state vector z_k capturing the complete computational history at step k, with each component serving a specific cryptographic purpose as summarized in Table II [28].The indices f_k and b_k track the current frame and block position respectively, while V_k^{orig} and V_k^{mod} accumulate commitments to the original and modified video streams, and F_k^{orig} and F_k^{mod} serve as the corresponding frame-level accumulators. The key insight is that by embedding both the intra-frame spatial accumulator (F_k^x) and the inter-frame temporal accumulator (V_k^x) within the same state vector, the architecture enables the folding scheme to seamlessly switch between intra- and inter-frame processing without maintaining separate proof chains.

A Boolean control governs the coordination between the inner and outer loops' signal L_k, which indicates whether the current block is the last in its frame. This signal is not a free prover input but is hard-constrained within the circuit as $L_k \equiv (b_k = B - 1)$, where B is the verifier-fixed block count committed at state slot $z[6]$, preventing the prover from triggering premature or delayed frame termination. This signal serves a dual purpose, computationally determining which

state components to update and cryptographically controlling the recursive structure of the folding operation. When $L_k = 0$, the system operates in inner-loop mode, indicating that more blocks remain in the current frame. When $L_k = 1$, the system transitions to outer-loop mode, finalizing the current frame and preparing for the next. This switching mechanism maps directly onto Nova's folding scheme [17]: inner-loop steps fold block-level R1CS instances into the growing frame accumulator F_k^x, while outer-loop steps fold the completed frame into the video accumulator V_k^x. The state transitions, illustrated in Fig. 2, enable the system to maintain temporal coherence while processing video data incrementally. When $L_k = 0$ (inner-loop transition), the system processes a single block BlockData_k with temporal buffer T_k through the verifiable computational module (such as the grayscale, cropping, spatiotemporal CNN, and Vision Transformer processors [42], [43]), updating only the frame accumulator F_k^x while preserving inter-frame context, where h_k^{feat} denotes the block feature hash and $x \in \{\text{orig}, \text{mod}\}$ is a stream indicator in which $x = \text{orig}$ refers to the original video stream and $x = \text{mod}$ refers to the modified (processed) video stream:

$$\begin{cases} f_{k+1} = f_k, \quad b_{k+1} = b_k + 1 \\ V_{k+1}^x = V_k^x \\ F_{k+1}^x = \text{Poseidon}(F_k^x,\, h_k^{\text{feat}},\, b_k,\, t_k) \\ T_{k+1} = \mathcal{U}_{\text{in}}(T_k,\, h_k^{\text{feat}}) \end{cases} \tag{5}$$

The frame index f_k remains unchanged, reflecting that processing continues within the same frame. The block index b_k increments, tracking spatial progress. Critically, the video accumulator V_k^x remains frozen, as no frame has been finalized. The frame accumulator F_k^x updates via Poseidon hashing, cryptographically binding the new block's feature hash, its spatial position b_k, and timestamp t_k into the growing commitment. At the genesis block ($f_k = 0, b_k = 0$), the circuit enforces $\delta_{\text{start}} \cdot (h_1^{\text{rlc}} - \tau_{\text{C2PA}}) = 0$. To circumvent the prohibitive in-circuit overhead of standard C2PA SHA-256 signatures [29], our framework adopts a Commit-and-Prove approach [44] in which the manifest is authenticated off-circuit. This allows τ_{C2PA} to be directly instantiated as the ZK-friendly Poseidon hash [35] of the verified genesis pixels, serving as a public input to cryptographically anchor the proof chain. The temporal buffer T_k updates through the module-specific function $\mathcal{U}_{\text{in}}$, which implements different behaviors depending on the computational module. For stateless operations like grayscale conversion, $\mathcal{U}_{\text{in}}$ acts as the identity function, simply preserving T_k. For temporal CNN operations, $\mathcal{U}_{\text{in}}$ accumulates spatial features into the sliding window buffer to enable cross-frame inference [45]. For Vision Transformer modules, $\mathcal{U}_{\text{in}}$ updates the cross-block context buffer, propagating intra-frame sequential dependencies between consecutive blocks. This design allows for the same circuit $\mathcal{F}$ to support multiple video processing tasks via parameterized state-update functions.

When $L_k = 1$ (outer-loop transition, as shown in the rightmost step of Fig. 2), the system finalizes the current frame and advances to the next, updating the video accumulator and resetting the frame accumulator :

$$\begin{cases} F_k'^x = \text{Poseidon}(F_k^x,\, h_k^{\text{feat}},\, b_k,\, t_k) \\ f_{k+1} = f_k + 1, \quad b_{k+1} = 0 \\ V_{k+1}^x = \text{Poseidon}(V_k^x,\, F_k'^x,\, f_k,\, t_k) \\ F_{k+1}^x = 0 \\ T_{k+1} = \mathcal{U}_{\text{out}}(T_k,\, F_k'^x) \end{cases} \tag{6}$$

The first step computes $F_k'^x$ by folding in the final block's contribution, producing a complete cryptographic commitment to all B blocks in frame f_k. The frame index f_k increments, signaling the transition to a new frame. The block index b_k resets to zero, preparing for spatial processing of the next frame. The video accumulator V_{k+1}^x is obtained by hashing the current accumulator V_k^x with the finalized frame commitment $F_k'^x$, frame index f_k, and timestamp t_k. This creates a cryptographic chain where V_{k+1}^x depends on V_k^x, which depends on V_{k-1}^x, and so on, forming an immutable temporal sequence. The frame accumulator F_{k+1}^x resets to zero, discarding the spatial history since it has been compressed into V_{k+1}^x. Finally, the temporal buffer updates via $\mathcal{U}_{\text{out}}$, which implements frame-level aggregation. For CNN modules, $\mathcal{U}_{\text{out}}$ shifts the sliding window forward, evicting the oldest frame's features and incorporating $F_k'^x$ as the newest entry. For cropping or grayscale modules that lack temporal dependencies, $\mathcal{U}_{\text{out}}$ simply preserves the existing buffer. For ViT modules, $\mathcal{U}_{\text{out}}$ resets the cross-block context buffer to zero upon frame completion, preparing for the next frame's sequential block processing.

This dual-mode mechanism maps precisely onto Nova's folding scheme [17], [45]. Each inner-loop step corresponds to one folding operation that combines the current accumulated frame state (represented by F_k^x) with a new block's R1CS instance. The witness vector for this folding includes the block's pixel data, module-specific intermediate activations (convolution and ReLU outputs for the CNN module, or patch embeddings and attention outputs for the ViT module), and the current temporal buffer T_k. Each outer-loop step corresponds to a higher-level folding operation that combines the accumulated video state (V_k^x) with the finalized frame commitment ($F_k'^x$), where $x \in \{\text{orig}, \text{mod}\}$. The hierarchical Poseidon-based hash commitment aggregation [35] ensures that both levels maintain the cryptographic binding necessary for soundness. Tampering with any block invalidates F_k^x, which in turn invalidates V_k^x, causing the final proof to fail verification.

The video step transformer $\mathcal{F}_T : (\mathbb{Z}_p^{11} \times \Theta) \to \mathbb{Z}_p^{11}$ (where $d = 11$ and Θ denotes private inputs defined in Table II) enforces three critical constraints within the R1CS circuit. First, the temporal consistency constraint $\mathcal{C}_t$ ensures strict monotonic timestamp progression. For all $k > 0$, we require $t_k > t_{k-1}$. Proper boundary handling at $k = 0$ sets the initial timestamp to a reference value t_{ref}. This constraint is encoded as an R1CS inequality check that prevents frame reordering or replay attacks. Second, the boundary safety constraint $\mathcal{C}_b$ (instantiated as the Boundary Safety Zone in Fig. 1) validates that intermediate computational values remain within stage-specific certified bit-widths w_k, preventing overflow attacks

where adversaries craft malicious inputs to cause arithmetic wraparound in the finite field $\mathbb{Z}_p$. Formally, $\mathcal{C}_b^{(k)}$ decomposes each intermediate value ν at processing stage k into w_k bits and verifies:

$$\mathcal{C}_b^{(k)} : \nu = \sum_{i=0}^{w_k-1} b_i \cdot 2^i \tag{7}$$

where each $b_i \in \{0,1\}$ is verified as an R1CS constraint. Specifically, an 8-bit width is adopted for raw pixel inputs to constrain them within the $[0, 255]$ range. For the spatial convolution outputs, an 11-bit width is assigned based on the extreme boundary produced when applying the Laplacian kernel to 8-bit inputs, yielding a theoretical maximum absolute value of 2040, which falls within the representable range of $2^{11} = 2048$. The temporal convolution outputs are set to a 12-bit width to safely accommodate the accumulation of spatial and historical features across frames (bounded within $2^{12} = 4096$). For Vision Transformer intermediate values, a 16-bit width is adopted to accommodate the wider dynamic range produced by multi-head attention projections and feed-forward layers. Beyond bit-width enforcement, $\mathcal{C}_b$ additionally hard-constrains the loop control signal L_k, ensuring that frame termination occurs exactly at the declared boundary, and enforces the C2PA genesis binding at the first block of the first frame. This stage-specific certification ensures that all intermediate values remain within their physically valid numeric ranges throughout the computation, closing the attack vector whereby a malicious prover could exploit finite-field wraparound to produce a valid proof on physically impossible inputs.

The memory efficiency of this architecture arises from a fundamental property of the Two-Level IVC design: frame-level states are compressed into constant-size commitments before processing the next frame. Formally, after processing frame f, the system maintains only the dual video accumulators V_f^{orig} and V_f^{mod} (two field elements) tracking the original and modified video streams respectively, and the temporal buffer T_f, whose full feature vector is never stored in the IVC state. Only its fixed-size Poseidon hash commitment occupies the corresponding state slot, preserving the $O(1)$ memory guarantee irrespective of feature dimensionality. The frame accumulator F_f^x is discarded after being compressed into V_f^x, and all B block-level witnesses are similarly discarded after being compressed into F_f^x, where $x \in \{\text{orig}, \text{mod}\}$.

This design achieves constant $O(1)$ memory complexity with respect to video length F. At any point during processing of frame k, the system stores only the current state vector of fixed size $d = 11$ field elements, the current block's circuit witness whose size is determined solely by the block configuration S, and the temporal buffer whose size remains fixed regardless of k. In contrast, traditional flat recursive schemes must maintain the entire computational history, leading to an $O(B \cdot F)$ memory requirement as more frames are processed [19], whereas our architecture maintains a per-frame memory footprint that is strictly independent of video duration. The hierarchical structure further improves proof time scalability. Inner-loop folding operations have a constant per-step cost determined solely by the block size S, independent of the total number of frames, while outer-loop operations remain constant-size regardless of how many blocks were processed in the current frame. This separation of concerns allows the architecture to scale gracefully with both spatial resolution and temporal length.

B. Pluggable Verifiable Computational Modules

The primary innovation of our Two-Level IVC architecture is its modularity, achieved through the Verifiable Computational Module (VCM). Defined as a tuple $\mathcal{M} = (g, \mathcal{K}, \mathcal{W})$ [44], where g denotes the computation function, $\mathcal{K}$ the constraint system, and $\mathcal{W}$ the witness generator, a VCM decouples specific video operations from the recursive engine. We demonstrate this flexibility by progressing systematically from fundamental pixel transformations to geometric manipulations, spatiotemporal convolution, and finally to Vision Transformer inference.

We begin with the Grayscale Integrity Processor to establish the baseline pattern for pixel-level verification. Algorithm 1 implements both the grayscale and cropping operations within a unified pixel-level VCM, distinguished by the parameter $op \in \{\text{GRAYSCALE}, \text{CROP}\}$. For grayscale conversion ($op1 = \text{GRAYSCALE}$), the module enforces the standard luminosity transform by calculating a weighted sum of RGB channels and performing integer division within the circuit. Range constraints ensure the output `gray` remains within the valid 8-bit interval $[0, 255]$, preventing finite-field wraparound. The block-level commitment is computed as a RLC hash $h_k^{\text{rlc}} = \sum_{i=0}^{P-1} \text{pack}(r_i, g_i, b_i) \cdot \rho_k^i$, where ρ_k is a Fiat-Shamir challenge derived from the current IVC state, binding all P pixels of the block into a single field element before Poseidon accumulation [35]. At the genesis block, the module additionally enforces $\delta_{\text{start}} \cdot (h_1^{\text{rlc}} - \tau_{\text{C2PA}}) = 0$, binding the first block's RLC commitment to the camera-attested C2PA credential.

To extend verification to geometric consistency, the Secure Cropping Processor is instantiated as $op2 = \text{CROP}$ within the same Algorithm 1. This module validates spatial mappings via selective block retention. For each block, a binary retention flag $is\_kept \in \{0, 1\}$ indicates whether the block originates from the authorized crop window $\mathcal{W}$. The modified accumulator F^{mod} is updated only when $is\_kept = 1$, realized by a MUX gate in the circuit. This ensures that blocks outside the authorized region do not influence the accumulated proof commitment. Beyond pixel-level and geometric operations, the Spatiotemporal CNN Processor (Algorithm 2) extends the framework to verifiable deep learning inference by mapping neural operations to R1CS constraints while managing temporal dependencies. Intra-frame spatial features are extracted by decomposing convolutions into vector products and linearizing ReLU activations. The 4×4 patch size exceeds the 3×3 Laplacian kernel dimensions, ensuring that each patch is self-contained and no pixel data from neighboring blocks is accessed. Beyond spatial self-containment, for inter-frame continuity, the module retrieves the historical feature buffer H_{t-1}^{eff} (zero-initialized at genesis as enforced by the canonical state constraint in Equation (10)) from the IVC state

Algorithm 1 Two-Level IVC: Grayscale Conversion and Secure Cropping Step Circuit (Unified Pixel-Level VCM)

Require: $\{(r_i, g_i, b_i)\}_{i=1}^{P}$; $op \in \{\text{GRAYSCALE}, \text{CROP}\}$; $is\_kept \in \{0,1\}$ (if Crop);
$\mathbf{z}=(f_k, b_k, V^{\text{orig}}, F^{\text{orig}}, V^{\text{mod}}, F^{\text{mod}}, B, \cdot, \cdot, ts_k, \tau_{\text{C2PA}})$; ρ; ts
Ensure: Updated state $\mathbf{z}'$
// Step 1: Shared Security Constraints
Enforce8Bit$(r_i, g_i, b_i)\ \forall i$
$\delta_{\text{start}} \leftarrow \text{IsZero}(f_k) \wedge \text{IsZero}(b_k)$
Enforce : $\delta_{\text{start}} \cdot (\text{RLC}(\{r_i, g_i, b_i\}, \rho) - \tau_{\text{C2PA}}) = 0$ ▷ C2PA anchor
Enforce : $(1 - \delta_{\text{start}}) \cdot (ts \le ts_k) = 0$ ▷ Temporal monotonicity
// Step 2: Module-Specific Transformation
$rlc\_orig \leftarrow \text{RLC}(r_i + g_i \cdot 2^{10} + b_i \cdot 2^{20},\ \rho)$
if $op = \text{GRAYSCALE}$ **then**
  Enforce : $1000 \cdot gray_i + rem_i = 299 r_i + 587 g_i + 114 b_i\ \forall i$
  Enforce8Bit$(gray_i)$
  $rlc\_mod \leftarrow \text{RLC}(gray_i \cdot (1 + 2^{10} + 2^{20}),\ \rho)$
else
  $rlc\_mod \leftarrow rlc\_orig$ ▷ Crop reuses original RLC
end if
// Step 3: Dual-Chain Frame Accumulation
$F^{\text{orig}} \leftarrow \text{Poseidon}(F^{\text{orig}},\ rlc\_orig,\ b_k,\ ts)$
if $op = \text{GRAYSCALE}$ **then**
  $F^{\text{mod}} \leftarrow \text{Poseidon}(F^{\text{mod}},\ rlc\_mod,\ b_k,\ ts)$
else
  $F^{\text{mod}} \leftarrow \text{MUX}\big(is\_kept,\ \text{Poseidon}(F^{\text{mod}}, rlc\_orig, b_k, ts),\ F^{\text{mod}}\big)$
end if
// Step 4: Outer-Loop Boundary
if $b_k = B - 1$ **then**
  $F'^{x} \leftarrow \text{Poseidon}(F^{x},\ rlc,\ b_k,\ ts),\ x \in \{\text{orig}, \text{mod}\}$
  $V^{x} \leftarrow \text{Poseidon}(V^{x}, F'^{x}, f_k,\ ts),\ x \in \{\text{orig}, \text{mod}\}$
  $f_k \mathrel{+}= 1;\quad b_k \leftarrow 0;\quad ts_k \leftarrow ts$
else
  $b_k \mathrel{+}= 1;\quad ts_k \leftarrow ts$
end if
return $\mathbf{z}'$

Algorithm 2 Two-Level IVC: Spatiotemporal CNN Step Circuit

Require: Sampled patches P_t; prover history $H_{\text{prv}}, H_{\text{prv},2}$;
IVC state $\mathbf{z}=(f_k,\ b_k,\ V,\ F,\ h_m,\ T,\ B,\ H_{t-1},\ \cdot,\ ts,\ \tau_{\text{C2PA}})$;
weights Θ_s, Θ_t
Ensure: Updated state $\mathbf{z}'$
EnforceRange$(p_{i,j}, 8)\ \forall p_{i,j} \in P_t$
$\delta_{\text{start}} \leftarrow \text{is\_zero}(f_k) \wedge \text{is\_zero}(b_k)$
Enforce : $H_{t-1}^{\text{eff}} = \text{cond\_sel}(\delta_{\text{start}}, \mathbf{0}, H_{\text{prv}})$ ▷ Zero-init at genesis
Enforce : $\delta_{\text{start}} \cdot (\text{Poseidon}(P_t) - \tau_{\text{C2PA}}) = 0$ ▷ C2PA anchor
for each sampled patch $p \in P_t$ **do**
  $S_p \leftarrow \text{SafeMax}(\text{ZK-ReLU}(p \cdot \Theta_s, 11), 11)$
end for
$T_t \leftarrow \text{ZK-ReLU}(\Theta_{t,0} \cdot S_t + \Theta_{t,1} \cdot H_{t-1}^{\text{eff}} + \Theta_{t,2} \cdot H_{\text{prv},2},\ 12)$
$F \leftarrow \text{Poseidon}(F,\ \text{Poseidon}(T_t),\ b_k,\ ts)$
if is_last **then**
  $V \leftarrow \text{Poseidon}(V, F, f_k,\ ts)$
  $f_k \mathrel{+}= 1;\quad b_k \leftarrow 0$
else
  $b_k \mathrel{+}= 1$
end if
return $\mathbf{z}'$

and fuses it with the current spatial features S_t via temporal convolution, before updating the buffer to serve as context for the next frame. This sliding-window design ensures the proof cryptographically covers the semantic evolution of the video stream across all frames [42], [43].

The framework further extends to the Vision Transformer (ViT) Processor (Algorithm 3) [46]. Because monolithic full-frame processing is computationally infeasible on commodity hardware, our architecture first partitions frames into blocks yielding L patches ($H_p \times W_p \times C$). This hardware-adaptive parameter L (SEQ_LEN) trades per-step circuit size for faster overall proof generation. During the layer-by-layer proof generation, processing these patches yields high-dimensional intermediate tensors at each step. To transfer the cryptographic commitments derived from these tensors to the next layer while maintaining a constant $O(1)$ memory footprint within the IVC state vector, we introduce HierHash$(\cdot)$, a two-stage compression mechanism based on the Poseidon hash function. It hashes a 2D tensor into a 1D array row by row, and then absorbs this 1D array chunk by chunk into a running accumulator chain to yield a single cryptographic commitment.

To accommodate zero knowledge proof constraints, specific algebraic optimizations are applied during circuit construction. The Softmax layer uses a nine segment piecewise linear approximation of $\exp(-x)$. GeLU is replaced by the constraint efficient ReLU [47], [48], and standard deviation scaling simplifies to mean centering normalization following established ZK ML paradigms [49]. The inner loop processes each ViT block through $N_{\text{layers}} = 12$ consecutive IVC steps (one Transformer layer per step), distributing the forward pass without materializing the full activation tensor. The IVC state vector increments the layer index l per step. The block counter b_k advances only upon completing the final layer ($l = 11$). The outer loop finalizes the frame level commitment $F_k'^{x}$ after each block and updates the video level accumulator V_k^{x} at verified frame boundaries, preserving an $O(1)$ peak memory guarantee irrespective of video duration or block count.

Notably, this modular substitution of the VCM does not affect the outer recursive structure of the Two-Level IVC framework. The outer Poseidon accumulation chains, timestamp monotonicity checks, and C2PA trust root anchoring remain unchanged across VCM transitions. All VCM modules operate within the same 11-dimensional state vector $z_k \in \mathbb{Z}_p^{11}$, where each state slot carries a constant-size field element regardless of the underlying computation. The semantic assignment of individual state slots varies according to module-specific temporal requirements. In the spatiotemporal CNN module, $z[7]$ carries the inter-frame historical hash accumulator H_{t-1}^{eff}, enabling cross-frame temporal inference. In the ViT module, $z[7]$ carries a Poseidon hash commitment to the cross-block context buffer of dimension D_{model}, propagating intra-frame sequential dependencies between consecutive blocks. For both stateful modules, only the hash commitment of the full feature vector occupies the corresponding state slot, preserving the $O(1)$ peak memory guarantee irrespective of feature dimensionality or video duration. For stateless modules such as grayscale conversion and cropping, $z[7]$ defaults to zero throughout the proof chain.

C. Security and Privacy Guarantees

We analyze the architecture's security based on the hardness of the Discrete Logarithm Problem and the binding properties of Pedersen commitments [17], [50]. For any polynomial-time adversary $\mathcal{A}$ producing a valid proof π, there exists an efficient

Algorithm 3 Two-Level IVC: Layer-Granular ViT Step Circuit

Require: $\mathbf{z}=(f_k, b_k, V, F, h_m, ts, ctx, l, ca\_hash, \tau_{\text{C2PA}})$; $W_{\text{emb}}, W_1^{(l)}, W_2^{(l)}$; P_t; is_last
Ensure: Updated state $\mathbf{z}'$
1: $\text{EnforceRange}(p_{i,j}, 8)\ \forall p_{i,j}$; $\text{Enforce}(h_m = \mathbf{z}[4])$
2: Enforce : $\delta_{\text{start}} \cdot (\text{HierHash}(P_t) - \tau_{\text{C2PA}}) = 0$ ▷ C2PA anchor
3: Enforce : $(l{>}0) \cdot (\text{HierHash}(\text{CachedAct}) - ca\_hash) = 0$
4: Enforce : $(b_k{>}0 \wedge l{=}0) \cdot (\text{Hash}(ctx) - \mathbf{z}[7]) = 0$
5: $X \leftarrow \lfloor W_{\text{emb}} \cdot P_t / 1024 \rfloor$
6: $H_{\text{in}} \leftarrow \text{MUX}(l{=}0, \text{Fuse}(X, ctx), \text{CachedAct})$
7: $H_1 \leftarrow \text{MHA}(\text{LN}(H_{\text{in}})) + H_{\text{in}}$
8: $H_{\text{out}} \leftarrow \lfloor W_2^{(l)} \cdot \text{ZK-ReLU}(\lfloor W_1^{(l)} \cdot \text{LN}(H_1)/1024 \rfloor)/1024 \rfloor + H_1$
9: $il \leftarrow (l{=}N_{\text{layers}}{-}1)$; $sil \leftarrow (il \wedge is\_last)$
10: $F' \leftarrow \text{MUX}\big(il,$ $\text{Poseidon}(F, \text{HierHash}(H_{\text{out}}), \text{HierHash}(P_t)),\ F\big)$
11: $V' \leftarrow \text{MUX}(sil,\ \text{Poseidon}(V, F', f_k,\ ts),\ V)$
12: $f_k' \leftarrow \text{MUX}(sil, f_k{+}1, f_k)$; $l' \leftarrow \text{MUX}(il, 0, l{+}1)$
13: $b_k' \leftarrow \text{MUX}(sil, 0, \text{MUX}(il, b_k{+}1, b_k))$
14: $ctx' \leftarrow \text{MUX}(sil, 0, \text{MUX}(il, \text{Hash}(H_{\text{out}}[\text{SEQ\_LEN}{-}1]), ctx))$
15: **return** $\mathbf{z}'$

extractor $\mathcal{E}$ recovering a valid witness w' with overwhelming probability:

$$\Pr[\mathcal{R}(w') = 1 \mid \mathcal{V}(\text{vk}, x, \pi) = 1, w' \leftarrow \mathcal{E}(\pi)] \geq 1 - \epsilon(\lambda) \quad (8)$$

where $\mathcal{R}(w') = 1$ denotes that witness w' satisfies the R1CS circuit constraints, and $\epsilon(\lambda)$ is negligible. The inner loop encodes block-level computations (pixel transformations, CNN operations, and Vision Transformer inference) as R1CS constraints, while the outer loop aggregates them via the folding scheme in Section III-A. The Pedersen binding property ensures the witness z_k cannot be altered without breaking the discrete logarithm assumption, forming an immutable chain across all frames.

As summarized in Table IV, our framework achieves all five security properties. Static image verification systems such as VIMz [19] and VerITAS [20] operate on individual frames without temporal continuity, and thus cannot provide temporal integrity guarantees. Eva [21] achieves temporal integrity through a sequential hash chain, but its reliance on a local H.264 decoder for public input extraction exposes the verification perimeter to decoder supply-chain vulnerabilities. Additionally, its Schnorr signatures are incompatible with the standard C2PA ECDSA specification, precluding direct integration with existing C2PA-compliant devices. In contrast, our scheme enforces temporal ordering through explicit in-circuit frame index constraints and operates directly on raw pixels without codec dependencies, eliminating decoder supply-chain risks. Native C2PA compatibility is achieved through in-circuit anchor binding.

For stateful modules (CNN and ViT), the temporal buffer T_k acts as a cryptographic chaining variable binding frame k to frame $k-1$. For stateless modules, inter-frame binding is maintained solely through the video accumulator V_k^x. Any manipulation of frame ordering or content requires producing a valid witness referencing inconsistent historical states, which is computationally infeasible under the folding scheme's soundness guarantee.

We define the adversarial success event $\mathcal{E}_{\text{forge}}$ with security condition $\Pr[\mathcal{E}_{\text{forge}}] \leq \text{negl}(\lambda)$:

$$\mathcal{E}_{\text{forge}} := (\mathcal{V}(\pi) = 1) \wedge \begin{pmatrix} (\exists k > 0 : t_k \leq t_{k-1}) \\ \vee\ (\hat{\Theta} \neq \Theta) \\ \vee\ (z_0 \neq (\mathbf{0}_{d-1}, h_m)) \end{pmatrix} \quad (9)$$

This formulation defends against three attack vectors. Temporal ordering violation is prevented by $\mathcal{C}_t$ enforcing strict timestamp progression $t_k > t_{k-1}$ at every folding step. Parameter tampering is detected via $h_m = \text{Poseidon}(\Theta)$ committed in z_0 and propagated through all frames, where $\hat{\Theta}$ denotes the parameters actually used by the prover during proof generation. State replay is prevented by the canonical initialization:

$$z_0 = \big(\underbrace{0,0}_{f_0, b_0},\ \underbrace{0,0,0,0}_{V_0^{\text{orig}}, F_0^{\text{orig}}, V_0^{\text{mod}}, F_0^{\text{mod}}},\ \underbrace{B}_{z[6]},\ \underbrace{1,0}_{p_0, M_0},\ \underbrace{t_{\text{ref}}}_{t_0},\ \underbrace{\tau_{\text{C2PA}}}_{z[10]}\big) \quad (10)$$

where t_{ref} denotes the initial reference timestamp at the start of video capture. The verifier-fixed constant B at $z[6]$ binds the proof to a specific spatial resolution, and τ_{C2PA} at $z[10]$ binds it to a specific capture device, ensuring that any proof transplantation across video sequences produces an initial state mismatch. Block boundary transitions are further enforced via the hard constraint L_k, implemented as an R1CS equality at every IVC step. When $b_k < B - 1$, this constraint forces $L_k = 0$, preventing premature frame termination. When $b_k = B - 1$, it forces $L_k = 1$, triggering frame finalisation exactly at the declared boundary. Any attempt to redistribute blocks unevenly across frames therefore directly violates this constraint. The circuit additionally enforces zero-initialization of H_{t-1}^{eff} at genesis to prevent injection of arbitrary historical states:

$$H_{t-1}^{\text{eff}} = (1 - \delta_{\text{start}}) \cdot H_{\text{prv}} + \delta_{\text{start}} \cdot \mathbf{0} \quad (11)$$

making it computationally infeasible to inject a non-zero historical state without violating the soundness of R1CS [51].

The security guarantees differ across VCM modules according to their verification mechanisms. For pixel-level modules (grayscale and cropping), block integrity is enforced via the RLC commitment $h_k^{\text{rlc}} = \sum_{i=0}^{P-1} \text{pack}(r_i, g_i, b_i) \cdot \rho_k^i$, where the Fiat-Shamir challenge $\rho_k = \text{Poseidon}(\eta_k, t_k, F_k^{\text{orig}})$ is derived from the current IVC state. By the Schwartz-Zippel lemma, the collision probability is bounded by $P/|\mathbb{F}_p|$, which is negligible for practical field sizes. For the spatiotemporal CNN module specifically, the HR-PRS mechanism samples n receptive fields using the C2PA trust anchor $z_0[10]$ as the pseudo-random seed:

$$p_i = \text{Poseidon}(z_0[10], i) \pmod{D_f}, \quad i = 0, \ldots, n-1 \quad (12)$$

with the routing constraint enforced via a MUX structure:

$$sf_{p_i} = \sum_{j=0}^{D_f - 1} \mathbf{1}[j = p_i] \cdot \phi_j, \quad i = 0, \ldots, n-1 \quad (13)$$

where ϕ_j denotes the j-th element of the feature vector and D_f is the feature vector dimension. Since $z_0[10]$ is bound at the moment of physical capture, sampling coordinates are unpredictable to the adversary. Although the coordinates are

deterministically derived from τ_{C2PA}, under the pseudorandomness assumption of the Poseidon hash function [35], this construction is computationally indistinguishable from uniform random sampling, connecting the mechanism to the leakage-resilient computation framework [52], whereby partial verification over pseudorandomly selected positions achieves exponentially small adversarial success probability. The per-block evasion probability satisfies $P_{\text{evade}} \leq (1-\gamma)^n$, and the global bound across all F frames and B blocks is:

$$P_{\text{evade}}^{\text{total}} \leq (1-\gamma)^{n \cdot F \cdot B} \tag{14}$$

With $n = 90$, $\gamma = 0.15$, $F = 3$, $B = 66$, this collapses below 10^{-1258}, well below the cryptographic baseline of 2^{-128}. For the Vision Transformer module, integrity is guaranteed by the 12-step IVC chain covering the complete forward pass, where the layer index l is embedded in the state vector to prevent layer skipping, each layer's output is committed via a hierarchical Poseidon hash HierHash$(\cdot)$, and the C2PA genesis anchor applies identically to the CNN module. Beyond integrity, the proof π reveals no information about the witness w beyond the public outputs x. For any polynomial-time distinguisher $\mathcal{D}$, a simulator $\mathcal{S}$ produces indistinguishable proofs:

$$|\Pr[\mathcal{D}(\pi_{\text{real}}) = 1] - \Pr[\mathcal{D}(\pi_{\text{sim}}) = 1]| \leq \text{negl}(\lambda) \tag{15}$$

where $\pi_{\text{real}} \leftarrow \mathcal{P}(x, w)$ and $\pi_{\text{sim}} \leftarrow \mathcal{S}(x)$. This guarantees that verification does not compromise the confidentiality of raw pixel content or proprietary model parameters, a property inherited from the Nova folding scheme and Spartan compression [17], [19].

IV. Experiment and Performance Analysis

A. Experiments Setting

We conduct the experiments in a `VMware Workstation` virtual machine running `Ubuntu 22.04.5 LTS`. The virtual machine is configured with an 8-core Intel® Xeon® CPU E5-2673 v3 @ 2.40 GHz processor and 64 GB of allocated memory. Our framework operates without GPU acceleration. During proof generation, multi-threaded concurrent execution is fully activated, and all timing results reflect wall-clock elapsed time. For Eva reproduction, an NVIDIA GeForce RTX 2060 GPU (8 GB VRAM, CUDA 12.9) is used to replicate Eva's GPU-accelerated proof generation pipeline [21]. The system prototype is implemented in `Python 3.11.7` within an `Anaconda` environment, where `OpenCV 4.11.0` handles video reading and image preprocessing, `NumPy 2.3.3` provides numerical computation support, and `psutil 5.9.0` is used for system resource monitoring during benchmarking. The parameterized bit-width architecture assigns stage-specific certified bit-widths as detailed in Section III-B, with bit-decomposition constraints enforced within the circuit to prevent finite-field wraparound attacks. The test dataset includes multiple standard-resolution videos spanning resolutions from 480p (640×480) to 4K (3840×2160), with pixel values normalized to the 8-bit input range prior to circuit ingestion.

B. Performance Evaluation

Table III characterizes the block size trade-off for the grayscale verification module across 720p, 1080p, and 4K resolutions. For a fixed resolution, increasing the block size reduces proof generation time by amortizing per-step folding overhead over fewer IVC steps: at 720p, proof time decreases from 2,229.5 s at 8×8 to 68.9 s at 128×36, a $32\times$ reduction, while peak memory increases proportionally from 0.22 GB to 1.10 GB.

Verification time is governed by two competing factors: the per-step constraint count C, which drives the Inner-Product Argument (IPA) verifier complexity at $O(C)$, and the total IVC step count k, which determines the cost of recursive folding chain validation in Nova's transcript. While the growth in C is the dominant trend, causing verification time to rise from 0.080 s ($C = 3{,}895$) to 0.698 s ($C = 231{,}095$) at 720p, the reduction in k as block size increases occasionally offsets a portion of this cost, producing local non-monotonicity. For example, at the transition from 20×24 to 32×24, the constraint count increases by a factor of 1.58 while the step count decreases by a factor of 1.60; because these two ratios are comparable, the competing effects partially cancel, and verification time decreases slightly from 0.201 s to 0.175 s. A similar near-equilibrium occurs between 32×36 and 40×40, where the constraint ratio (1.38) and the step-count ratio (1.39) are again nearly matched, yielding verification times of 0.300 s and 0.298 s, respectively. Outside these operating points, the C-factor dominates, and verification time resumes its upward trajectory with block size.

Proof size scales logarithmically in C per the Spartan sumcheck transcript structure, growing from 26.60 KB to 29.83 KB despite a 59-fold increase in C, confirming that the near-constant proof size is a structural property of the compressed SNARK. Both verification time and proof size are independent of video duration for a fixed block configuration. The block size therefore serves as a hardware-adaptive tuning parameter. Selecting a larger block configuration within the available memory budget reduces proof generation time while incurring a proportional increase in peak memory, reflecting a classic performance–resource trade-off. Within the range of configurations evaluated in this work, the observed peak memory for each verification module remains well within the capacity of current commodity workstations.

Figure 3 compares our framework against VIMz, VerITAS, and Eva using a 40×40 block configuration across 480p to 4K resolutions for the grayscale operation. We position our framework as an offline, codec-agnostic video verification system targeting commodity CPU deployment; the performance differences discussed below reflect fundamental architectural trade-offs rather than implementation deficiencies. Our proof size remains near-constant at $\sim$29 KB across all resolutions, whereas VerITAS scales from 2,490 KB at 480p to over 10,000 KB at 4K due to its $O(P)$ proof complexity. Eva achieves the smallest proof size (0.45 KB) by exploiting H.264 codec structures, though its CP-SNARK decider stage incurs 37–42 GB of constant peak memory. Our peak memory remains below 4.31 GB across all tested configurations, within

TABLE III
PERFORMANCE COMPARISON OF BLOCK SIZE STRATEGIES ACROSS DIFFERENT RESOLUTIONS

Resolution	Block Size	Non-linear Constraints	Ratio (%)	Steps	Proof (s)	Verif. (s)	Mem. (GB)	Size (KB)
720p (1280×720, 921,600 pixels)								
720p	8 × 8	3,895	0.007	14,400	2,229.5	0.080	0.22	26.60
720p	16 × 16	13,495	0.028	3,600	561.8	0.110	0.24	27.47
720p	20 × 20	20,695	0.043	2,304	366.7	0.103	0.32	27.41
720p	20 × 24	24,695	0.052	1,920	316.1	0.201	0.36	28.25
720p	32 × 24	39,095	0.083	1,200	205.9	0.175	0.37	28.29
720p	40 × 20	40,695	0.087	1,152	198.8	0.189	0.38	28.22
720p	32 × 30	48,695	0.104	960	170.6	0.210	0.41	28.27
720p	32 × 36	58,295	0.125	800	149.5	0.300	0.45	29.05
720p	40 × 40	80,695	0.174	576	114.8	0.298	0.49	29.04
720p	64 × 36	115,895	0.250	400	91.2	0.389	0.71	29.06
720p	128 × 36	231,095	0.500	200	68.9	0.698	1.10	29.83
1080p (1920×1080, 2,073,600 pixels)								
1080p	48 × 24	58,295	0.056	1,800	417.9	0.305	0.51	29.07
1080p	40 × 40	80,695	0.077	1,296	347.7	0.291	0.51	29.09
1080p	48 × 36	87,095	0.083	1,200	333.7	0.244	0.53	29.07
1080p	48 × 40	96,695	0.093	1,080	327.0	0.381	0.64	29.09
1080p	60 × 40	120,695	0.116	864	314.2	0.578	0.87	29.89
4K (3840×2160, 8,294,400 pixels)								
4K	40 × 40	80,695	0.019	5,184	1,403.4	0.320	0.70	29.02
4K	80 × 80	320,695	0.077	1,296	927.7	1.240	1.60	30.69
4K	96 × 72	346,295	0.083	1,200	887.7	1.210	1.77	30.68
4K	120 × 90	540,695	0.130	768	709.1	2.020	2.73	31.50
4K	192 × 216	2,074,295	0.500	200	549.2	3.421	4.31	32.34

Notes: Non-linear Constraints follow the formula $C = 50P + 695$, where $P = W_b \times H_b$ denotes the block pixel count and 695 is a fixed overhead from Poseidon permutation gates, security constraints (timestamp monotonicity, C2PA trust root binding), and MUX gadgets. Ratio = Block Area / Total Pixels × 100%; Steps = total recursive folding steps ($F \times B_{\text{per\_frame}}$), where F denotes the number of frames and $B_{\text{per\_frame}}$ the number of blocks per frame under the given block configuration.

TABLE IV
SECURITY PROPERTY AND ARCHITECTURAL COMPARISON OF AUTHENTICATION SCHEMES

Scheme	Target Media	Hardware Trust Root	Temporal Integrity (Replay/Reorder)	Codec-independent Verification	Standard C2PA Compatibility	Edit Pipeline Privacy
VIMz	Image	✓	N/A	✓	N/A	✓
VerITAS	Image	✓	N/A	✓	✗[b]	✓
Eva	Video	✓	✓ (Implicit)[a]	✗[d]	✗[c]	✓
Ours	Video	✓	✓ (Explicit)	✓	✓	✓

[a] Implicit: integrity via hash chain; Explicit: enforced by in-circuit frame index constraints [21].
[b] VerITAS requires a non-standard lattice-based hash preprocessing scheme incompatible with current C2PA implementations [20].
[c] Eva adopts Schnorr signatures incompatible with the C2PA ECDSA specification, precluding direct integration with existing C2PA-compliant devices [21].
[d] Eva's verifier must locally run an H.264 decoder to extract intermediate data as public inputs. A compromised decoder could introduce blind spots within the verification perimeter [21].

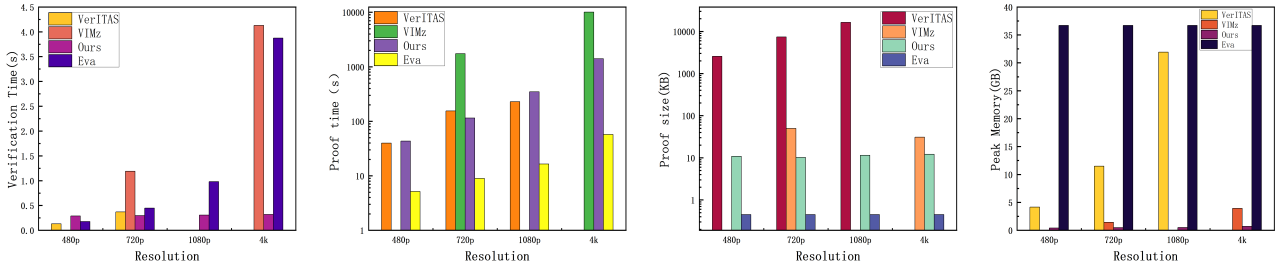

Fig. 3. Performance Comparison with State-of-the-Art Schemes. We compare our proposed framework against VIMz [19], VerITAS [20], and Eva [21] across four key metrics: (a) Verification Time, (b) Proof Time, (c) Proof Size, and (d) Peak Prover Memory. Our experiments use the 40 × 40 block configuration. Verification time for our scheme is invariant to video duration for a fixed block configuration, as both proof size and verification cost depend only on the per-step constraint count and the number of IVC steps per frame, not on the total frame count. Eva's verification time grows linearly with video length due to the verifier's independent H.264 decoding step [21].

commodity hardware capacity. Eva achieves the fastest per-frame proof generation through GPU acceleration and H.264-specific circuit optimization; VIMz exhibits the longest proof times due to its $O(N)$ row-wise processing complexity.Our verification time remains constant with respect to video duration, determined primarily by the per-step constraint count C. By contrast, as reported in [21], Eva's verification cost grows linearly with video length as the verifier must independently decode the entire bitstream prior to proof checking: 114.3 s for a 1-minute 720p sequence and 229.3 s for a 2-minute sequence.

Table V presents the cropping verification comparison. Under the 40×40 configuration, our system achieves consistent proof time of 303.6 s and peak memory of 0.48 GB regardless of output resolution, since circuit complexity depends solely on source resolution. Adopting larger block configurations further reduces proof time: the 192×108 configuration achieves 115.7 s at 2.70 GB peak memory for 1080p sources, the 128×144 configuration achieves 31.3 s at 2.81 GB for 720p sources, and the 144×96 configuration achieves 15.6 s at 1.46 GB for 480p sources. These results confirm that the block size parameter serves as an effective hardware-adaptive tuning knob. Selecting larger blocks within the available memory budget consistently reduces proof generation time by amortizing per-step folding overhead across fewer IVC steps. By contrast, VerITAS proof time and memory scale with output resolution (12.85–128.67 s, 1.99–13.71 GB), and VIMz supports only two fixed resolution pairs with substantially longer proof times (3,485.97–24,752.97 s). Eva achieves faster cropping proof times (5.88–19.35 s) through GPU acceleration and H.264 optimization, but requires 42.6 GB of constant peak memory and uses hardcoded crop coordinates that are not exposed as public circuit inputs, limiting flexibility for arbitrary cropping operations. Notably, our 480p configuration (144×96, 15.6 s, CPU-only) approaches Eva's 480p proving time (5.88 s, GPU-accelerated) without requiring GPU resources, demonstrating that the block-size mechanism effectively compensates for the absence of hardware acceleration. Our proof size remains constant at 29.04–31.53 KB across all tested configurations, compared to VerITAS which ranges from 2,490 to 15,990 KB.

The following scalability experiments uniformly adopt the 40×40 block configuration at 720p resolution as a controlled variable to isolate differences in computational complexity across operations from any artifacts introduced by varying block dimensions. Quantitative results validate the theoretical efficiency across three critical dimensions. First, peak memory remains stable across video lengths (Fig. 4), with approximately 0.46 GB for CNN, 0.49 GB for grayscale, and 0.41 GB for cropping, confirming that the outer loop effectively aggregates history without state explosion. Second, Fig. 5 shows a strict linear relationship between proof generation time and frame count. Cropping achieves the lowest per-frame proving time (111.9 s/frame), followed by grayscale (114.8 s/frame), while the spatiotemporal CNN requires the longest (336.4 s/frame) due to ZK-ReLU activations and temporal convolutions. The CNN circuit benefits from the HR-PRS sampling mechanism, which bounds the active constraint count to $n \times P_{\text{patch}} = 90 \times 16 = 1440$ regardless of block area, preventing super-linear growth in proof time with

TABLE V
PERFORMANCE COMPARISON FOR IMAGE CROPPING VERIFICATION

Method	Source→Output	Proof (s)	Verif. (s)	Mem. (GB)	Size (KB)
VerITAS [20]	Any→1080p	128.67	0.86	13.71	15990
	Any→720p	26.68	0.37	3.88	7180
	Any→480p	12.85	0.13	1.99	2490
VIMz [19]	4K→1080p	24752.97	3.88	9.58	56
	No 720p circuit available				
	720p→480p	3485.97	2.56	2.997	32
Eva [21]†	1080p→Cropped	19.35	1.032	42.6	0.45
	720p→Cropped	10.37	0.477	42.6	0.45
	480p→Cropped	5.88	0.2	42.6	0.45
Ours‡	1080p→1080p (40×40)	303.6	0.310	0.48	29.04
	1080p→720p (40×40)	303.6	0.271	0.48	29.04
	1080p→480p (40×40)	303.6	0.236	0.48	29.04
	1080p→1080p (192×108)	115.7	2.270	2.70	31.53
	720p→Any§ (128×144)	31.3	2.19	2.81	31.46
	480p→Any§ (144×96)	15.6	1.19	1.46	30.69

† Eva: hardcoded crop coordinates; crop region is not a public circuit input.
‡ Ours: verification time is invariant to output resolution and video duration for a fixed block configuration.
§ "Any" denotes an arbitrary crop region strictly smaller than the source resolution; upsampling to a higher resolution is not supported.
Circuit complexity: VerITAS (output size), VIMz (fixed pairs), Eva (source size), Ours (source size only).

TABLE VI
SYSTEM PERFORMANCE OPTIMIZATION ACROSS BATCH SIZES AND EXTENDED MULTI-FRAME VIDEO SEQUENCES

SEQ_LEN	Frames	Constraints	Steps	T_{prove} (h)	T_{verify} (s)	Size (KB)	RAM (GB)
1	1	183,505	16,200	2.15	0.74	11.34	0.95
2	1	347,848	8,100	1.53	1.36	11.64	1.75
4	1	692,706	4,056	1.25	2.50	11.95	3.05
8	1	1,445,638	2,028	1.04	5.14	12.25	5.94
16	1	3,204,366	1,020	**0.97**	**9.91**	**12.55**	11.73
16	2	3,204,366	2,040	1.94	**9.91**	**12.55**	11.73
16	5	3,204,366	5,100	4.85	**9.91**	**12.55**	11.73
16	10	3,204,366	10,200	9.70	**9.91**	**12.55**	11.73

Note: For $\text{SEQ\_LEN} = 16$, T_{verify} and proof size remain constant across all frame counts, with measurement variations within ± 0.03 s and ± 0.2 KB respectively.

increasing resolution. Finally, as shown in Figs. 6 and 7, both verification time (approximately 0.10–0.307 s) and proof size (approximately 10.73 KB for CNN and 29.04 KB for grayscale and cropping) remain constant with respect to video length due to the final Spartan compression.

Table VI evaluates the ViT verification module at 480p resolution across varying SEQ_LEN configurations and frame counts. Increasing SEQ_LEN from 1 to 16 reduces proof generation time from 2.15 h to 0.97 h by distributing the 12-layer Transformer forward pass over fewer IVC steps. This acceleration incurs a peak memory tradeoff, growing from 0.95 GB to 11.73 GB, confirming SEQ_LEN as the primary hardware-adaptive parameter for the ViT module. Under the $\text{SEQ\_LEN} = 16$ configuration, proof generation time scales linearly with the frame count (ranging from 0.97 h for 1 frame to 9.70 h for 10 frames), whereas the verification time (9.91 s) and proof size (12.55 KB) remain invariant.

For broader context, a contemporaneous Halo2-based system, ZK-APEX [53], verifies unlearning operations on a ViT-B/16 model, reporting approximately 2 h of proof generation time. However, the verification target of their system is restricted to linear operations, specifically executing matrix-vector products for weight updates based on offline-aggregated statistics derived from static, low-resolution (224×224) images. In contrast, our framework verifies the complete 12-layer ViT forward pass, which incorporates non-linear components including piecewise-linear activations, layer normalization, and multi-head attention, executed directly over continuous raw video streams at resolutions of 480p (640×480) and beyond. By processing these non-linear components to maintain a hardware-rooted C2PA provenance chain, our Two-Level IVC architecture achieves $O(1)$ proof sizes (12.55 KB) and

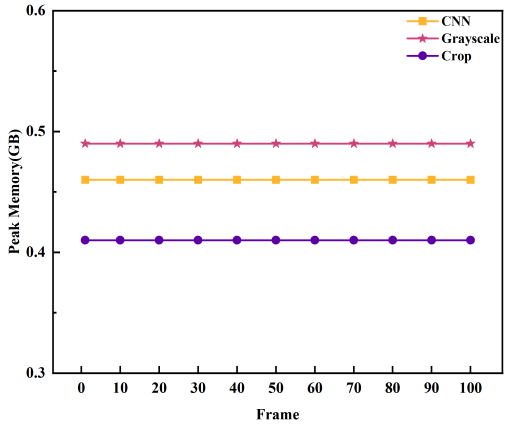

Fig. 4. Peak memory usage: Near-constant across all operations.

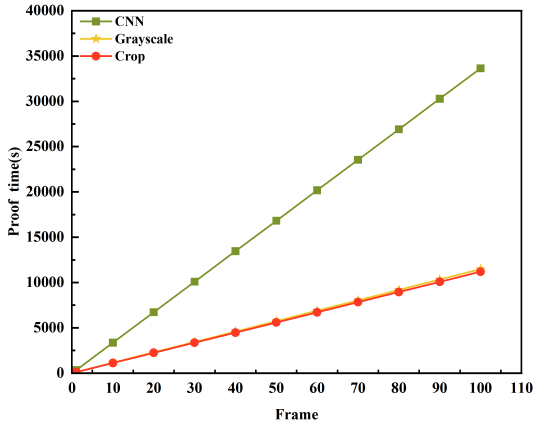

Fig. 5. Proof generation time scalability across video operations.

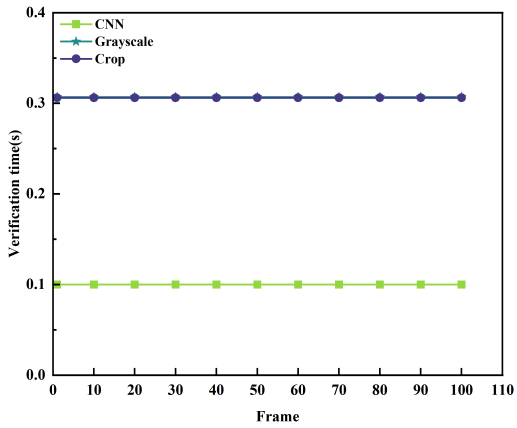

Fig. 6. Verification time: Constant overhead independent of frame count.

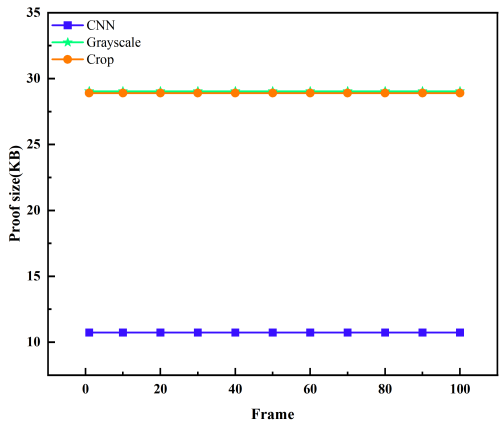

Fig. 7. Constant proof size across video lengths for different operations.

verification times (9.91 s), providing a structural comparison to the 400 MB proofs and 10 minutes verification times reported for the linear operations in ZK-APEX.

V. Conclusion

This paper proposes an end-to-end Two-Level Incremental Verifiable Computation (IVC) architecture to address the conflict between data privacy and computational trust in batch-processing intelligent video analysis. By decoupling the recursive verification mechanism from core computational logic, our framework successfully verifies essential spatiotemporal operations while strictly enforcing temporal integrity through cryptographic state propagation. Experimental results confirm that, unlike traditional monolithic SNARK schemes, where resource consumption scales linearly with input size, our approach achieves offline end-to-end integrity verification for high-definition video streams with constant $O(1)$ memory overhead with respect to video length F on commodity CPU hardware, demonstrating the feasibility of batch-processing arbitrarily long video sequences for high-stakes analytics (e.g., forensic audit, medical imaging review) without memory exhaustion.

In future work, we aim to pursue two primary directions. First, we plan to investigate hardware acceleration strategies, including GPU-assisted witness generation and parallel block proving to reduce proof generation time and move toward practical deployment on resource-constrained platforms. Second, we intend to extend the current VCM module suite to support more computationally demanding architectures such as Three-Dimensional Residual Networks (3D ResNets) and large-scale Transformer variants with increased numbers of attention layers, building on the ViT circuit foundation established in this work. Taken together, these efforts aim to broaden both the computational efficiency and the supported computational scope of the proposed verifiable video processing framework.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grants 62561035 and 62471205,

and in part by Yunnan Fundamental Research Projects under Grants 202601CI070104 and 202301AV070003. The authors extend their appreciation to the Deanship of Scientific Research and Graduate Studies at King Khalid University, KSA, for funding this work through the Small Research Group under grant number RGP.1/39/46.

REFERENCES

[1] N. Tang, C. Yang, J. Fan, L. Cao, Y. Luo, and A. Halevy, "VerifAI: Verified Generative AI," in *Proceedings of the 14th Conference on Innovative Data Systems Research (CIDR 2024)*, Chaminade, CA, USA, 2024, pp. 1–8.

[2] R. Zhu, D. Guo, D. Qi, Z. Chu, X. Yu, and S. Li, "A survey of trustworthy representation learning across domains," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 7, pp. 1–53, 2024.

[3] A. Malanowska, W. Mazurczyk, T. K. Araghi, D. Megías, and M. Kuribayashi, "Digital watermarking—a meta-survey and techniques for fake news detection," *IEEE Access*, vol. 12, pp. 36 311–36 345, 2024.

[4] X. Zhang, R. Li, J. Yu, Y. Xu, W. Li, and J. Zhang, "Editguard: Versatile image watermarking for tamper localization and copyright protection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11 964–11 974.

[5] A. Kaur, A. N. Hoshyar, V. Saikrishna, S. Firmin, and F. Xia, "Deepfake video detection: challenges and opportunities," *Artificial Intelligence Review*, vol. 57, no. 6, p. 159, 2024.

[6] X. Wu, X. Liao, B. Ou, Y. Liu, and Z. Qin, "Are Watermarks Bugs for Deepfake Detectors? Rethinking Proactive Forensics," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, 2024, pp. 6089–6097.

[7] W. Li, Z. Zhang, Z. Zhou, and Y. Deng, "A survey of concise non-interactive zero-knowledge proofs," *Journal of Cryptography*, vol. 9, no. 3, pp. 379–447, 2022.

[8] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von Neumann architecture," in *Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14)*, San Diego, CA, USA, 2014, pp. 781–796.

[9] B. S. Pranati, D. Suma, C. M. Latha, and S. Putheti, "RETRACTED CHAPTER: Large-Scale Video Classification with Convolutional Neural Networks," in *Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems*, 2020, pp. 689–695.

[10] B. Feng, Z. Wang, Y. Wang, S. Yang, and Y. Ding, "Zeno: A type-based optimization framework for zero knowledge neural network inference," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, 2024, pp. 450–464.

[11] H. Sun, T. Bai, J. Li, and H. Zhang, "Zkdl: Efficient zero-knowledge proofs of deep learning training," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 914–927, 2024.

[12] D. Kang, T. Hashimoto, I. Stoica, and Y. Sun, "Zk-img: Attested images via zero-knowledge proofs to fight disinformation," *arXiv preprint arXiv:2211.04775*, 2022.

[13] J. Groth, "On the size of pairing-based non-interactive arguments," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2016, pp. 305–326.

[14] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge," Cryptology ePrint Archive, Paper 2019/953, 2019. [Online]. Available: https://eprint.iacr.org/2019/953

[15] F. H. Soureshjani, M. Hall-Andersen, M. Jahanara, J. Kam, J. Gorzny, and M. Ahmadvand, "Automated Analysis of Halo2 Circuits," in *Proceedings of the 21st International Workshop on Satisfiability Modulo Theories (SMT 2023)*, ser. CEUR Workshop Proceedings, vol. 3429, Rome, Italy, 2023, pp. 1–16. [Online]. Available: https://ceur-ws.org/Vol-3429/paper3.pdf

[16] P. D. Monica, I. Visconti, A. Vitaletti, and M. Zecchini, "Trust nobody: Privacy-preserving proofs for edited photos with your laptop," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 4624–4642.

[17] A. Kothapalli, S. Setty, and I. Tzialla, "Nova: Recursive zero-knowledge arguments from folding schemes," in *Annual International Cryptology Conference*, 2022, pp. 359–388.

[18] P. Mallozzi, "Deploying zkp frameworks with real-world data: Challenges and proposed solutions," *arXiv preprint arXiv:2307.06408*, 2023.

[19] S. Dziembowski, S. Ebrahimi, and P. Hassanizadeh, "VIMz: Private Proofs of Image Manipulation using Folding-based zkSNARKs," *Proc. Priv. Enhanc. Technol.*, vol. 2025, no. 2, pp. 344–362, 2025.

[20] T. Datta, B. Chen, and D. Boneh, "VerITAS: Verifying image transformations at scale," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 4606–4623.

[21] C. Zhang, X. Yang, D. Oswald, M. Ryan, and P. Jovanovic, "Eva: Efficient Privacy-Preserving Proof of Authenticity for Lossily Encoded Videos," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 4643–4662.

[22] L. Du, A. T. S. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Signal Processing: Image Communication*, vol. 81, p. 115713, 2020.

[23] J. W. Seow, M. K. Lim, R. C. W. Phan, and J. K. Liu, "A comprehensive overview of Deepfake: Generation, detection, datasets, and opportunities," *Neurocomputing*, vol. 513, pp. 351–371, 2022.

[24] C. Zhang, S. Yu, Z. Tian, and J. J. Q. Yu, "Generative adversarial networks: A survey on attack and defense perspective," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–35, 2023.

[25] F. Khelifi and A. Bouridane, "Perceptual video hashing for content identification and authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 50–67, 2017.

[26] K. Li, R. Hong, J. Lang, J. Wu, F. Zhou, and J. Song, "MATCH: Multi-Agentic Evidence Grounding for Explainable Hate Video Detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 36, no. 7, pp. 9646–9659, 2026.

[27] S. Li, Z. Guo, C. Miao, W. Yang, L. Wang, G. Yang, and X. Liao, "Prototype Memory-based Neighboring Feature Fusion Network for Image Manipulation Localization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2026, DOI: 10.1109/TCSVT.2026.3679746.

[28] R. Lavin, X. Liu, H. Mohanty, L. Norman, G. Zaarour, and B. Krishnamachari, "A survey on the applications of zero-knowledge proofs," *arXiv preprint arXiv:2408.00243*, 2024.

[29] L. Rosenthol, "C2PA: the world's first industry standard for content provenance (conference presentation)," in *Applications of Digital Image Processing XLV*, vol. 12226, 2022, p. 122260P.

[30] G. Ramezan, E. R. Casas, B. Beath, and J. Godfrey, "zk-Database: Privacy-enabled Databases using Zero-Knowledge Proof," in *Proceedings of the 2024 7th International Conference on Blockchain Technology and Applications*, 2024, pp. 19–25.

[31] J. Lai, T. Wang, S. Zhang, Q. Yang, and S. C. Liew, "Biozero: An efficient and privacy-preserving decentralized biometric authentication protocol on open blockchain," *arXiv preprint arXiv:2409.17509*, 2024.

[32] S. Malik, N. Gupta, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "TradeChain: Decoupling traceability and identity in blockchain enabled supply chains," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2021, pp. 1141–1152.

[33] A. Kothapalli and S. Setty, "HyperNova: Recursive arguments for customizable constraint systems," in *Annual International Cryptology Conference*, 2024, pp. 345–379.

[34] C. Franck and J. Großschädl, "Efficient implementation of Pedersen commitments using twisted Edwards curves," in *International Conference on Mobile, Secure, and Programmable Networking*, 2017, pp. 1–17.

[35] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for Zero-Knowledge proof systems," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 519–535.

[36] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs, "Fiat-Shamir: from practice to theory," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 1082–1090.

[37] J. Sedlmeir, A. Rieger, T. Roth, and G. Fridgen, "Battling disinformation with cryptography," *Nature Machine Intelligence*, vol. 5, no. 10, pp. 1056–1057, 2023.

[38] M. Bellés-Muñoz, M. Isabel, J. L. Muñoz-Tapia, A. Rubio, and J. Baylina, "Circom: A circuit description language for building zero-knowledge applications," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4733–4751, 2022.

[39] N. Elias, "Lova: A Novel Framework for Verifying Mathematical Proofs with Incrementally Verifiable Computation," in *Proceedings of the 2025 ACM Workshop on Secure and Trustworthy Cyber-physical Systems*, 2025, pp. 63–72.

[40] A. Naveh and E. Tromer, "Photoproof: Cryptographic image authentication for any set of permissible transformations," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 255–271.

[41] M. Hall-Andersen and J. B. Nielsen, "On Valiant's conjecture: impossibility of incrementally verifiable computation from random oracles," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2023, pp. 438–469.
[42] T. Liu, X. Xie, and Y. Zhang, "Zkcnn: Zero knowledge proofs for convolutional neural network predictions and accuracy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2968–2985.
[43] B. Feng, L. Qin, Z. Zhang, Y. Ding, and S. Chu, "ZEN: Efficient Zero-Knowledge Proofs for Neural Networks," *IACR Cryptology ePrint Archive*, vol. 2021, p. 87, 2021.
[44] M. Campanelli, D. Fiore, and A. Querol, "LegoSNARK: Modular design and composition of succinct zero-knowledge proofs," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2075–2092.
[45] J. Liu, L. Guo, and T. Kang, "GENES: An Efficient Recursive zk-SNARK and Its Novel Application in Blockchain," *Electronics*, vol. 14, no. 3, p. 492, 2025.
[46] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*, 2021, pp. 10 347–10 357.
[47] M. Hao, H. Li, H. Chen, P. Xing, G. Xu, and T. Zhang, "Iron: Private inference on transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 718–15 731, 2022.
[48] Z. Peng, C. Zhao, T. Wang, G. Liao, Z. Lin, Y. Liu, B. Cao, L. Shi, Q. Yang, and S. Zhang, "A survey of zero-knowledge proof based verifiable machine learning," *Artificial Intelligence Review*, vol. 59, pp. 157–207, 2026.
[49] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "Zkml: An optimizing system for ml inference in zero-knowledge proofs," in *Proceedings of the Nineteenth European Conference on Computer Systems*, 2024, pp. 560–574.
[50] E. Morais, T. Koens, C. V. Wijk, and A. Koren, "A survey on zero knowledge range proofs and applications," *SN Applied Sciences*, vol. 1, no. 8, p. 946, 2019.
[51] Y. Zheng, Y. Cao, and C.-H. Chang, "A PUF-based data-device hash for tampered image detection and source camera identification," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 620–634, 2020.
[52] D. Genkin, Y. Ishai, and M. Weiss, "How to construct a leakage-resilient (stateless) trusted party," in *Theory of Cryptography Conference*, 2017, pp. 209–244.
[53] M. M. Maheri, S. Cotterill, A. Davidson, and H. Haddadi, "ZK-APEX: Zero-Knowledge Approximate Personalized Unlearning with Executable Proofs," in *Proceedings of the 9th Conference on Machine Learning and Systems (MLSys)*, Bellevue, WA, USA, 2026, pp. 1–21.

Fenhua Bai Fenhua Bai received the Ph.D. degree from the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China, in 2023. She is currently a lecturer with Kunming University of Science and Technology, Kunming, China. Her research interests include blockchains, edge computing, network security, and games application. E-mail: baifenhua@kust.edu.cn.

Xianlin Yang Xianlin Yang received his bachelor's degree from the School of Mechanical and Electrical Engineering, Sanming University, Sanming, China, in 2020. He is currently pursuing an M.S. degree at Kunming University of Science and Technology. His current research interests include privacy protection and zero-knowledge proofs. E-mail: yangxianlin@stu.kust.edu.cn.

Tao Shen Tao Shen received the Ph.D. degree in electrical engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2013. He is currently a Professor with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. His research interests include intelligent perception and computation, artificial intelligence, blockchain, and industrial Internet. E-mail: shentao@kust.edu.cn.

Bei Gong Bei Gong (Member, IEEE) received the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. He has published at least 14 national invention patents and over 100 academic articles in the past five years. Most of them are published in high-quality journals and conferences, such as IEEE TII, IEEE TSC, IEEE Internet of Things Journal, etc., which are SCI/EI and internationally renowned journals and conferences. He has presided over eight national projects, such as the National Key R&D Program, the National Natural Science Foundation of China, and the Major Special Project of Science and Technology, etc. His research interests include trusted computing, Internet of Things security, mobile Internet of Things, and mobile edge computing.

Muhammad Waqas Mahmoud Wagdy (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2010. From October 2019 to February 2021, he was a Postdoctoral Research Fellow with the Department of Information Technology, Beijing University of Information Engineering, Beijing. Since April 2022, he has been an Associate Professor with the Faculty of Engineering and Science, University of Bahrain, Bahrain. He is currently a Senior Lecturer with the Faculty of Engineering and Science, University of Greenwich, London, U.K. He has also been an Adjunct Senior Lecturer with the School of Computing, Edith Cowan University, Australia. He has published more than 150 peer-reviewed articles in high-impact journals, reputable conferences, covering vehicle networks, cybersecurity, and machine learning. He is recognized as Global Talent in the area of wireless communications by Stanford University (2022–2023) and Digital Systems. He is also a Guest Editor of Journal of Computing and Applied Sciences (MDPI).

Hisham Alfuhaid Hisham Alfuhaid (Member, IEEE) received the M.S. degree in computer science from George Washington University, Washington, DC, USA, in 2010. He is an Assistant Professor with the Department of Information Technology, King Khalid University, Abha, Saudi Arabia, since 2020, and he serves as the director of the Department of Information Technology at King Khalid University. His research interests include cloud computing, network security, and adversarial machine learning.