MotiMem: Motion-Aware Approximate Memory for Energy-Efficient Neural Perception in Autonomous Vehicles

Haohua Que[1,†], Mingkai Liu[2,†], Jiayue Xie[3], Haojia Gao[3], Jiajun Sun[3], Hongyi Xu[3,4], Handong Yao[1,*], Fei Qiao[3,*]

***Abstract*—High-resolution sensors are critical for robust autonomous perception but impose a severe "memory wall" on battery-constrained electric vehicles. In these systems, data movement energy often outweighs computation. Traditional image compression is ill-suited as it is semantically blind and optimizes for storage rather than bus switching activity. We propose MotiMem, a hardware-software co-designed interface. Exploiting temporal coherence, MotiMem uses lightweight 2D Motion Propagation to dynamically identify Regions of Interest (RoI). Complementing this, a Hybrid Sparsity-Aware Coding scheme leverages adaptive inversion and truncation to induce bit-level sparsity. Extensive experiments across nuScenes, Waymo, and KITTI with 16 detection models demonstrate that MotiMem reduces memory-interface dynamic energy by $\approx 43\%$ while retaining $\approx 93\%$ of the object detection accuracy, establishing a new Pareto frontier significantly superior to standard codecs like JPEG and WebP.**

***Index Terms*—Autonomous Vehicles, Memory-Constrained Learning, Hardware-Software Co-design, Embedded Vision**

I. Introduction

In the rapidly evolving field of autonomous vehicles (AVs) and mobile robotics, visual perception serves as the cornerstone of safety and decision-making. To detect distant obstacles and ensure reliable navigation, modern perception systems of AVs are increasingly equipped with high-resolution cameras (e.g., 4K or 8K) [5], [6]. However, this pursuit of visual fidelity imposes a severe penalty on the underlying hardware. The massive influx of high-dimensional visual data creates a significant bottleneck in memory bandwidth and system energy consumption [7], [8]. For battery-constrained edge devices, the energy cost of data movement and storage often exceeds that of computation itself [9], [10], exacerbating the well-known "Memory Wall" problem [11], [12]. Consequently, designing energy-efficient perception systems without compromising detection accuracy has become a critical challenge [13].

[1]University of Georgia; [2]Peking University; [3]Tsinghua University; [4]Infinity Robotics; [†]Co-first authors: Haohua Que and Mingkai Liu. [*]Corresponding authors: Handong Yao (Handong.Yao@uga.edu) and Fei Qiao (qiaofei@tsinghua.edu.cn). This work was supported by the Beijing Natural Science Foundation (L253009); the National Natural Science Foundation of China (62334006, U25A20489); the Brain Science and Brain-like Intelligence Technology National Science and Technology Major Project of China (Grant No. 2025ZD0215600); and the National Key Technologies R&D Program of China (2025YFF1500600). Haohua Que and Handong Yao completed their work entirely in the United States and received no funding support.

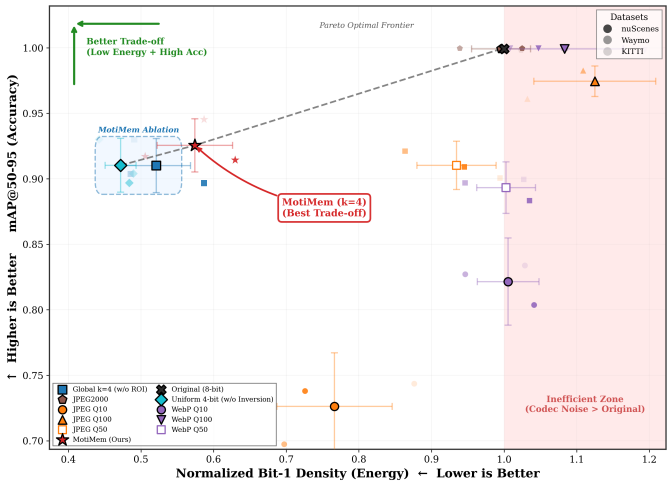

Fig. 1: **Pareto Efficiency Analysis (Accuracy vs. Energy Proxy) across nuScenes, Waymo, and KITTI Datasets [1]–[4].** MotiMem (marked with a red star) establishes a superior Pareto frontier compared to standard codecs (JPEG, WebP) which fall into the inefficient zone due to high entropy. Evaluated across **three diverse datasets** and **16 detection models**, MotiMem achieves the optimal trade-off by reducing the **normalized bit-1 density** (a direct proxy for dynamic energy) by $\approx 43\%$ while maintaining $\approx 93\%$ of the original uncompressed mAP. Crucially, MotiMem significantly outperforms the semantically blind "Global k=4" ablation (blue square), validating that **semantic-aware allocation** is superior to uniform quantization for energy-constrained perception.

To mitigate this memory pressure, conventional approaches typically resort to standard image compression standards (e.g., JPEG, WebP). While effective in reducing data volume, these methods are ill-suited for the "sensing-for-perception" pipeline for two fundamental reasons. First, they are *semantically blind*: they treat irrelevant background pixels (e.g., sky, road surface) with the same importance as safety-critical foreground objects, wasting precious bandwidth on non-informative content [14], [15]. Second, and more critically from a hardware perspective, standard codecs optimize solely for storage capacity rather than **bit-level activity** [16], [17]. The resulting compressed bitstreams often exhibit high entropy (randomness), which maximizes the switching activity factor (α) across **both off-chip buses and RAM**. [18], [19]. This leads to excessive

dynamic power dissipation during data transfer, negating the energy benefits of reduced volume [20], [21]. Furthermore, complex decoding processes introduce additional latency, which is detrimental to real-time control loops [22], [23].

To bridge the gap between high-fidelity sensors and energy-efficient neural networks, we propose **MotiMem**, a hardware-software co-designed neural interface [24], [25]. Unlike passive data pipes, MotiMem actively reshapes the sensor stream based on the *temporal coherence* inherent in driving scenarios [26], [27]. Our key insight is that physical objects do not move randomly; their continuous motion allows us to utilize detection results from the previous frame $(T-1)$ as a low-cost motion prior to predict Regions of Interest (RoI) for the current frame (T) [28]. Complementing this kinematic awareness, we introduce a ***Hybrid Sparsity-Aware Coding*** scheme. To maximize bus and memory efficiency, this approach differentially encodes data based on semantic importance: for critical RoIs, it applies **high-bit inversion** with a Least Significant Bit (LSB) flag to maintain full precision; for the background, it couples inversion with aggressive **low-bit truncation** [29], [30]. This hybrid strategy induces high bit-level sparsity [31], which directly minimizes switching activity across both the sensor-memory bus and the RAM arrays themselves [16], [17]. This allows the system to preserve target fidelity while significantly reducing memory-interface energy for robust neural-based autonomous perception (e.g., object detection). Recent advances in other high-stakes learning domains also reinforce this design intuition: structure-preserving graph modeling and multi-scale temporal fusion improve predictive robustness when task-relevant spatial/temporal cues are retained [32], [33].

Our main contributions are:

- We propose **MotiMem**, a hardware–software co-designed neural interface that leverages temporal coherence for non-uniform precision storage. It identifies RoIs via kinematic-guided motion propagation, avoiding the latency of heavy motion estimation.
- We introduce a ***Hybrid Sparsity-Aware Coding*** scheme that couples adaptive inversion with background truncation. It reduces the **normalized bitstream activity** (proxy for memory-interface dynamic energy) by $\approx 43\%$ (Fig. 1) while preserving task-critical fidelity.
- We conduct extensive experiments on video object detection across **nuScenes, Waymo, and KITTI** [1]–[4] and **16 detection models**, demonstrating a superior accuracy–energy Pareto frontier (Fig. 1).

II. Related Work

A. Perception Stream Compression

Standard codecs (e.g., JPEG, WebP) are primarily optimized for *Human Visual System (HVS)* fidelity (e.g., PSNR/SSIM) rather than downstream machine vision accuracy [34], [35]. They rely on block-based prediction/transform coding with quantization that suppresses high-frequency details, which can distort edge and texture cues important to CNNs [34], [36]. Critically, such approaches typically disregard the **hardware energy cost** of data movement [37], [38]. Moreover, many hardware-friendly compression/quantization schemes are *semantically blind* [39], [40], allocating uniform precision to both irrelevant background (e.g., sky) and safety-critical foreground objects. While learning-based codecs (e.g., VAE-based) can achieve superior rate–distortion performance [41]–[43], their neural decoding often incurs **prohibitive latency and compute/energy overhead** on edge hardware [44], [45]. Therefore, we compare against industry-standard codecs and hardware-friendly quantization baselines that represent the current low-latency operating point in automotive perception [46]–[49]. In contrast, **MotiMem** introduces a power-centric, RoI-guided coding paradigm: rather than only reducing bandwidth, it reduces the density of bit-1s and the **effective transition activity** in the storage stream, inducing high sparsity without necessarily reducing raw interface bandwidth. This directly lowers the **(weighted) switching activity** and thus dynamic power dissipation in RAM and bus interfaces [50], [51].

B. Perception-Aware RoI Prioritization

Different from codec-centric compression, these works prioritize *where* to preserve information for perception tasks. Recent studies on perception-aware visual systems investigate how to prioritize task-relevant regions to improve downstream robustness under constrained resources. Such methods range from semantic/object-centric region selection to RoI-driven quality allocation and foveated processing, especially in autonomous driving and networked perception settings [14], [15], [29], [30]. Nevertheless, two limitations hinder their deployment in real-time AV pipelines. First, RoI estimation is often coupled with additional computation (e.g., motion estimation or multi-stage semantic modules), which increases latency and complicates system integration [22], [23]. Second, these approaches typically optimize rate–accuracy (or rate–distortion) objectives, without explicitly targeting *memory-interface energy* metrics such as switching activity during storage and transfer [37], [38], [50], [51]. MotiMem addresses these limitations by deriving RoIs from prior detections via lightweight motion propagation and by shaping the stored bitstream to reduce memory-interface dynamic power.

C. Energy-Aware Neural Interfaces and Approximate Memory

Energy-efficient neural perception increasingly recognizes that the system-level energy is often dominated by memory access and data movement rather than arithmetic [37], [38]. At the interface level, dynamic power on buses and memory I/O is strongly governed by switching activity and data-dependent bit transitions, motivating transition-aware encoding schemes [50], [51]. Recent low-power memory designs further exploit bit-level manipulation, e.g., selective bit dropping combined with encoding co-strategies for DRAM/SRAM, to reduce switching and I/O energy [52]–[55]. In parallel, approximate computing and low-precision inference reduce storage/compute bit-width with controlled accuracy loss [39], [40],

[49]. However, existing methods are often task-agnostic or semantically uniform, applying approximation/encoding without accounting for which regions are critical to neural perception. MotiMem bridges this gap by co-designing motion-guided RoI semantics with sparsity-aware bitstream shaping (inversion and background truncation), directly targeting memory-interface dynamic power while preserving task-critical accuracy.

III. Methodology

A. Problem Setup and System Goal

We study a streaming on-device perception pipeline for autonomous driving. At each time step t, a high-resolution camera frame $I_t \in \mathbb{R}^{H\times W\times C}$ is captured and stored in memory before being consumed by a detector/tracker running on an on-device compute unit (CPU/NPU). The memory path (sensor $\rightarrow$ bus $\rightarrow$ RAM) is often energy-critical due to data-dependent activity on buses and memory writes. Unlike codec-centric designs that optimize only storage size, MotiMem targets **system-level energy** by shaping the *bit statistics* of the representation stored/transferred along the memory path, while maintaining detection/tracking quality. MotiMem takes the current frame I_t and the previous detection results D_{t-1} (bounding boxes, classes, confidence scores) as input. It outputs an encoded stream stored in RAM, decoded with lightweight bit operations, and fed into standard perception models.

B. Closed-Loop Overview

MotiMem consists of two coupled components (Fig. 2): (1) **Temporal Coherence $\rightarrow$ RoI Prediction:** leverage temporal coherence to predict a binary RoI mask M_t for the current frame using the prior detections D_{t-1}. (2) **RoI-Guided Bitstream Shaping:** use M_t to apply a hybrid coding strategy where RoI is coded with high fidelity (but not strictly lossless), and the background is aggressively simplified. Both paths apply bit-level shaping to bias the stored stream toward fewer 1s and fewer toggles. The key system intuition is that only a small portion of each frame is critical for perception, and thus representation fidelity can be allocated non-uniformly.

C. Temporal Coherence and RoI Prediction

As shown in Fig. 3, MotiMem predicts the RoI mask M_t for the current frame I_t by propagating prior detections D_{t-1} using lightweight 2D motion estimation. Let the detector output at the previous frame be

$$D_{t-1} = \{(b_i, c_i, \gamma_i)\}_{i=1}^{N_{t-1}}, \tag{1}$$

where b_i is a bounding box, c_i is class, and γ_i is confidence.

a) Lightweight 2D motion propagation.: Instead of optical flow, we propagate each box in the image plane using a constant-velocity model:

$$\hat{b}_i^{(t)} = b_i^{(t-1)} + \Delta t \cdot \mathbf{v}_i, \tag{2}$$

where $\mathbf{v}_i$ is a lightweight 2D velocity estimate (e.g., from recent box center displacement or a simple tracker). To tolerate uncertainty, we inflate the predicted box by margin δ:

$$\tilde{b}_i^{(t)} = \mathrm{Inflate}(\hat{b}_i^{(t)}, \delta). \tag{3}$$

b) RoI mask as control metadata (low overhead).: We build a binary RoI mask by the union of inflated boxes:

$$M_t(p) = \mathbb{I}\Big[p \in \bigcup_{i=1}^{N_{t-1}} \tilde{b}_i^{(t)}\Big], \quad p \in \{1,\dots,H\}\times\{1,\dots,W\}. \tag{4}$$

In practice, M_t is represented at block granularity (e.g., 16×16 blocks), so it is compact control metadata that does *not* traverse the high-volume pixel stream and introduces negligible overhead relative to I_t.

D. RoI-Guided Hybrid Coding with a Single Parameter k

We encode the current frame I_t into a shaped representation by routing pixels into two paths controlled by M_t, as shown in Fig. 4. Our design is inspired by low-power bit dropping/encoding strategies for memory interfaces [52], [53], and extends them to *neural-perception-aware* coding via RoI-guided non-uniform fidelity.

We use a single integer parameter k to control both: (i) background approximation strength by retaining only the top-k MSBs, and (ii) RoI bit shaping by selectively inverting the top-k MSBs, while using the *pixel LSB as an embedded inversion flag*. This choice keeps the stored pixel width unchanged (B bits per pixel/channel), avoiding raw interface bandwidth expansion.

Let $x_{t,p} \in \{0,\dots,2^B-1\}$ denote a B-bit pixel/channel value at location p. We decompose x into MSBs and the least-significant bit (LSB):

$$x = 2\cdot\bar{x} + \ell(x), \tag{5}$$

where $\ell(x)\in\{0,1\}$ is the LSB and $\bar{x}\in\{0,\dots,2^{B-1}-1\}$ contains the upper $(B-1)$ bits.

Define the top-k MSB mask over the full B-bit word:

$$\mathbf{m}^{(k)} \triangleq \sum_{j=B-k}^{B-1} 2^j. \tag{6}$$

1) RoI path: high-fidelity selective inversion with LSB-embedded flag: For RoI pixels ($M_t(p)=1$), we preserve perceptual fidelity while shaping bit statistics. We decide whether to invert the top-k MSBs based on their Hamming weight:

$$\eta_k(x) \triangleq \mathbb{I}\Big[w_H\big(x\wedge\mathbf{m}^{(k)}\big) > \tau\Big], \qquad \tau\in[0,k], \tag{7}$$

where we typically use $\tau = k/2$.

a) Flag embedding.: We embed the inversion decision into the LSB, i.e., we overwrite the LSB with the flag:

$$f(p) \triangleq \eta_k(x), \qquad \ell(x_{\text{enc}}^{\text{RoI}}) = f(p). \tag{8}$$

b) Selective inversion of MSBs.: We invert only the top-k bits when $f(p)=1$:

$$x_{\text{enc}}^{\text{RoI}} = \Big(x \oplus \big(f(p)\cdot\mathbf{m}^{(k)}\big)\Big) \text{ with LSB overwritten by } f(p). \tag{9}$$

This operation is *near-lossless*: only the original LSB may be lost because it is reused as the embedded flag, while the inverted MSB range is reversible given $f(p)$.

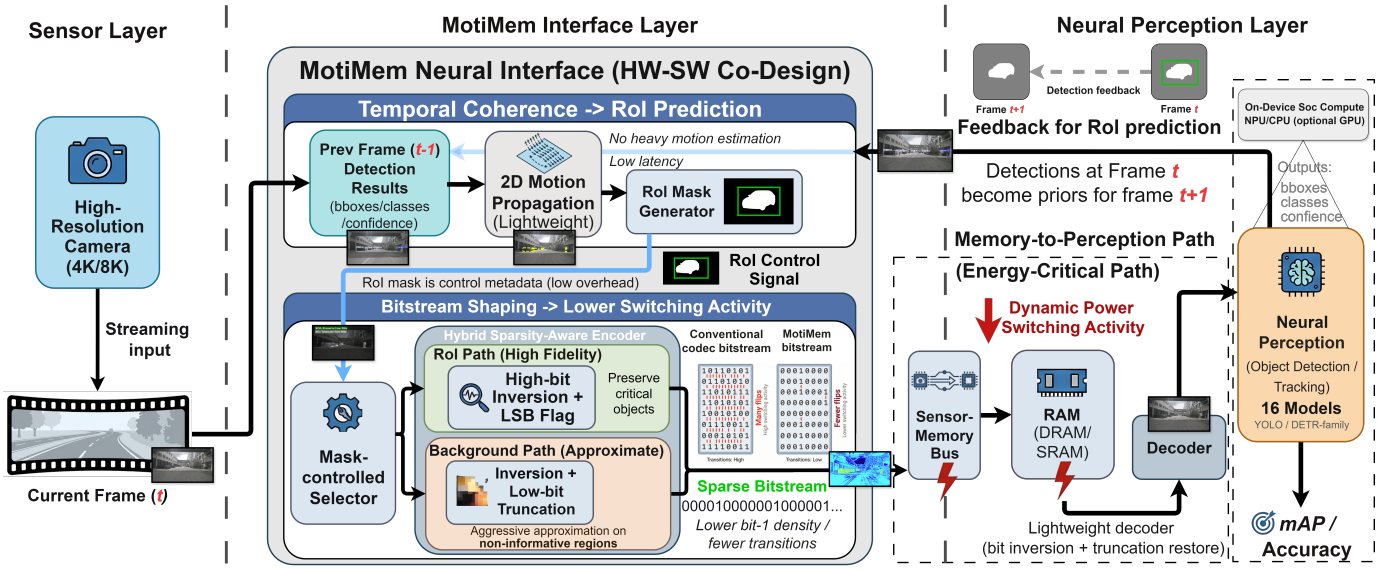

Fig. 2: **The MotiMem closed-loop interface for neural perception.** MotiMem sits between the camera stream and the memory hierarchy. It forms a closed loop: detections at time t are fed back as compact metadata to predict the RoI mask for time $t{+}1$. Guided by the RoI mask, MotiMem applies RoI-guided hybrid coding to shape the stored bitstream toward lower activity (fewer 1s/toggles) along the sensor-to-memory path, while preserving perception accuracy.

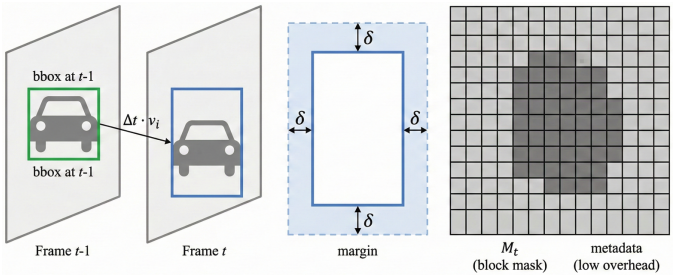

Fig. 3: **RoI prediction from temporal coherence.** Detections at frame $t{-}1$ are propagated to frame t using a lightweight 2D motion prior, inflated by margin δ, and rasterized to a compact block-level RoI mask M_t (e.g., 16×16 blocks).

c) Lightweight decoding.: At the decoder, we read the embedded flag from the LSB and invert back the same MSB range:

$$\hat{f}(p) = \ell(x_{\text{enc}}^{\text{RoI}}), \qquad \hat{x} = x_{\text{enc}}^{\text{RoI}} \oplus \big(\hat{f}(p) \cdot \mathbf{m}^{(k)}\big). \tag{10}$$

Because the original LSB was overwritten, $\hat{x}$ matches x up to a 1-bit error:

$$|x - \hat{x}| \le 1, \tag{11}$$

which empirically preserves detection accuracy (Fig. 1).

2) Background path: k-MSB truncation + MSB inversion with LSB-embedded flag: For background pixels ($M_t(p) = 0$), MotiMem enforces sparsity by a fixed two-step transform: **truncate** the low bits and **shape** the retained MSBs via selective inversion. The inversion decision is embedded into the pixel LSB as a flag, so the stream width remains B bits per pixel/channel.

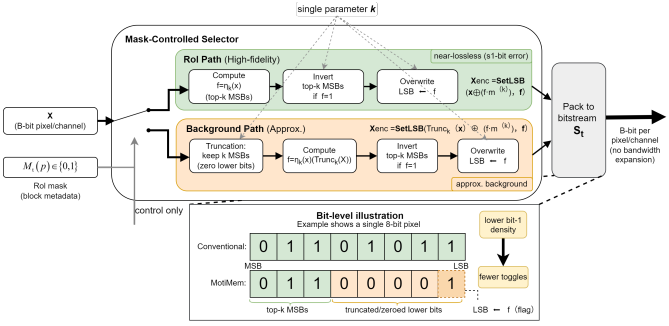

Fig. 4: **RoI-guided hybrid coding.** Pixels are routed into two paths based on $M_t(p)$ and parameter k. **RoI:** Compute and embed an inversion flag f in the LSB, selectively invert top-k MSBs if $f{=}1$. **Background:** Truncate to top-k MSBs, apply the same inversion, and embed f in the LSB. Both paths preserve B-bit width, reducing bit-1 density and transitions for lower memory energy.

a) Truncate low bits.: We keep only the top-k MSBs and zero the remaining $(B-k)$ bits:

$$x^{(k)} \triangleq \text{Trunc}_k(x) = \left\lfloor \frac{x}{2^{B-k}} \right\rfloor \cdot 2^{B-k}. \tag{12}$$

b) Invert dense MSB patterns and write the flag into LSB.: We compute an inversion indicator from the truncated value

$$f(p) \triangleq \eta_k\Big(x^{(k)}\Big) = \mathbb{I}\Big[w_H\big(x^{(k)} \wedge \mathbf{m}^{(k)}\big) > \tau\Big], \tag{13}$$

and apply inversion on the same top-k MSB range:

$$\tilde{x}^{(k)} \triangleq x^{(k)} \oplus \big(f(p) \cdot \mathbf{m}^{(k)}\big). \tag{14}$$

Finally, we overwrite the pixel LSB with the flag:

$$x_{\text{enc}}^{\text{BG}} \triangleq \text{SetLSB}\big(\tilde{x}^{(k)},\, f(p)\big), \qquad \ell\big(x_{\text{enc}}^{\text{BG}}\big) = f(p). \tag{15}$$

This background coding is intentionally approximate due to truncation (and LSB reuse), but it produces a sparse, low-activity stream that reduces normalized bit-1 density and toggle rate along the memory path.

3) Mask-controlled routing and bitstream packing: The per-pixel routing is:

$$x_{\text{enc}}(p) = \begin{cases} x_{\text{enc}}^{\text{RoI}}(p), & M_t(p) = 1, \\ x_{\text{enc}}^{\text{BG}}(p), & M_t(p) = 0. \end{cases} \tag{16}$$

The resulting encoded values are stored/transferred as a standard B-bit-per-pixel stream $\mathbf{s}_t$ (same tensor shape as I_t). Importantly, the inversion decision is embedded into the pixel LSB in *both* paths, so the representation width is unchanged and no additional raw interface bandwidth is introduced. The RoI mask M_t is compact control metadata (block-level) used only to steer routing/encoding and does not traverse the high-volume pixel stream. In our implementation, M_t is generated locally at the interface from D_{t-1} and thus does not require transmitting per-pixel side information.

E. Bitstream Activity and Energy Proxy Metrics

MotiMem reduces system energy by lowering the *data-dependent activity* of the stored/transferred bitstream on the sensor–memory path. We report two proxies [52], [53].

a) Normalized Bit-1 density.: For a frame-level bitstream $\mathbf{s} \in \{0,1\}^L$,

$$\rho_1(\mathbf{s}) \triangleq \frac{1}{L}\sum_{\ell=1}^{L} s_\ell, \qquad \text{NBD}_t \triangleq \frac{\rho_1(\mathbf{s}_t^{\text{enc}})}{\rho_1(\mathbf{s}_t^{\text{raw}})}. \tag{17}$$

Smaller NBD_t means fewer "1" bits in the encoded stream (higher sparsity), directly reducing dynamic energy.

b) Bit-transition activity: Partition $\mathbf{s}$ into W-bit words $\{\mathbf{s}_n\}_{n=1}^{N}$. The normalized transition rate is

$$\alpha(\mathbf{s}) \triangleq \frac{1}{(N-1)W}\sum_{n=2}^{N} d_H(\mathbf{s}_n, \mathbf{s}_{n-1}), \tag{18}$$

where $d_H(\cdot,\cdot)$ is the Hamming distance.

c) Effect of shaping: Truncation (Eq. (12)) forces most low-order bits to zero (except the LSB flag), and selective MSB inversion (Eq. (9), (15)) suppresses dense-one MSB patterns. Together, they reduce $\rho_1(\mathbf{s}^{\text{enc}})$ and $\alpha(\mathbf{s}^{\text{enc}})$, while using only lightweight bitwise operations.

F. Decoding and Closed-Loop Inference

Finally, the SoC performs lightweight decoding with only bitwise operations. For both RoI and background pixels, the decoder reads the embedded flag from the pixel LSB and reverses the selective top-k MSB inversion (Eq. (10)). For background pixels, the lower bits remain truncated (i.e., zeroed except the LSB used as the flag), yielding an approximate reconstruction. The decoded frame is then processed by standard detectors/trackers (e.g., YOLO/DETR-family), and the resulting detections are fed back as D_t to generate M_{t+1}, closing the loop. This feedback forms a closed-loop interface where perception outputs at t act as control metadata for encoding at $t{+}1$.

IV. Experimental Evaluation

A. Experimental Setup

a) Datasets & Models: **nuScenes** [1], **Waymo** [2], and **KITTI** [3], [4]. To ensure robustness, we report average performance across **16 diverse object detection models**, including the YOLO family (v5, v8, v9, v10, v11, YOLO26) and Transformer-based RT-DETR.

b) Baselines: (1) **Standard Codecs:** Industry standards including *JPEG* and *WebP* (configured at Q10, Q50, Q100), and lossless *JPEG2000*; (2) **Ablations:** To isolate component contributions, we evaluate *Global $k = 4$ (w/o ROI)* (applying uniform truncation to the whole frame) and *Uniform 4-bit (w/o Inversion)* (disabling the adaptive bit-flip optimization); (3) **Original (8-bit):** The uncompressed raw baseline, serving as the upper bound for accuracy.

c) Metrics: We report performance on two axes: (1) **Perception Accuracy**, measured by standard Mean Average Precision (**mAP@50-95**); and (2) **Energy Efficiency**, quantified by the **Normalized Bit-1 Density Ratio**. As defined in Eq. 17, this metric serves as a linear proxy for dynamic switching power in memory interfaces, where a ratio below 1.0 indicates effective energy reduction compared to the uncompressed baseline.

d) Implementation Details: The MotiMem encoding logic and detection pipeline were **functionally simulated** in software using PyTorch. Critically, D_{t-1} is obtained from **actual model detections** on the previously encoded frame (not ground-truth), faithfully emulating the closed-loop deployment. All experiments were conducted on a workstation with an Intel Core Ultra 9 285K CPU, 128GB RAM, and an NVIDIA RTX 6000 Pro GPU.

B. Sensitivity Analysis and Design Space Exploration

To determine the optimal operating point for the MotiMem interface, we analyze the sensitivity of both perception robustness and energy cost to the retained bit-width parameter k. We perform a sweep from $k = 1$ to $k = 7$ across the combined dataset. Fig. 5 illustrates these opposing trends.

a) The Energy Trend (Green Line): The green dashed line in Fig. 5 illustrates the energy cost. As we retain more bits ($k > 4$), the Normalized Bit-1 Density rises strictly monotonically. Specifically, increasing k from 4 to 7 raises the density ratio from ≈ 0.58 to ≈ 0.78 (a $\approx 35\%$ increase). This confirms that the lower bits in natural images are dominated by high-entropy noise. Including them contributes significantly to bus switching activity (cost) without providing meaningful structural information, as evidenced by the plateauing accuracy (Blue Line).

b) The Perception Trend (Blue Line): The blue solid line represents the average confidence score of the detection models, which serves as a proxy for feature integrity. We observe a law of diminishing returns: The accuracy exhibits two distinct

TABLE I: Comprehensive comparison of encoding methods across three autonomous driving datasets. All metrics are averaged over 16 detection models. MotiMem achieves the **optimal trade-off**: highest mAP retention among energy-efficient methods while consuming only 57% of the original energy.

Method	nuScenes					Waymo					KITTI					Average				
	mAP	Eng.	SSIM	PSNR	LPIPS	mAP	Eng.	SSIM	PSNR	LPIPS	mAP	Eng.	SSIM	PSNR	LPIPS	mAP	Eng.	SSIM	PSNR	LPIPS
Lossless Baselines																				
Original (8-bit)	1.00	1.00	1.00	∞	.000	1.00	1.00	1.00	∞	.000	1.00	1.00	1.00	∞	.000	1.00	1.00	1.00	∞	.000
JPEG2000	1.00	1.02	1.00	∞	.000	1.00	.94	1.00	∞	.000	1.00	1.02	1.00	∞	.000	1.00	1.00	1.00	∞	.000
MotiMem Variants (Ours)																				
MotiMem*	.91	.63	.92	30.6	.215	.92	.51	.93	31.8	.198	.95	.59	.91	29.4	.228	**.93**	**.57**	.92	30.6	.214
Global k=4	.90	.59	.92	30.2	.230	.90	.49	.93	31.4	.212	.93	.49	.91	29.0	.243	.91	.52	.92	30.2	.228
Uniform 4-bit	.90	.48	.91	29.5	.231	.90	.49	.92	30.6	.214	.93	.44	.90	28.3	.245	.91	.47	.91	29.5	.230
Standard Codecs																				
JPEG Q10	.74	.73	.85	31.7	.389	.70	.70	.86	32.9	.372	.74	.88	.84	30.5	.401	.73	.77	.85	31.7	.387
JPEG Q50	.91	.95	.97	39.7	.088	.92	.86	.97	41.2	.076	.90	.99	.96	38.4	.098	.91	.93	.97	39.8	.087
JPEG Q100	.98	1.23	1.00	54.0	.003	.98	1.11	1.00	55.8	.002	.96	1.03	1.00	52.6	.004	.97	1.12	1.00	54.1	.003
WebP Q10	.80	1.04	.88	33.8	.327	.83	.95	.89	35.1	.308	.83	1.03	.87	32.6	.342	.82	1.01	.88	33.8	.326
WebP Q50	.88	1.03	.94	37.4	.182	.90	.95	.95	38.8	.165	.90	1.03	.93	36.1	.196	.89	1.00	.94	37.4	.181
WebP Q100	1.00	1.05	.99	47.1	.017	1.00	1.01	1.00	48.6	.014	1.00	1.19	.99	45.8	.021	1.00	1.08	.99	47.2	.017

***Pareto optimal**: MotiMem is the only method achieving both high mAP (>0.92) and low energy (<0.6).
In Average column: blue = best, red = worst among lossy methods. Eng. = Normalized Bit-1 Density (↓).

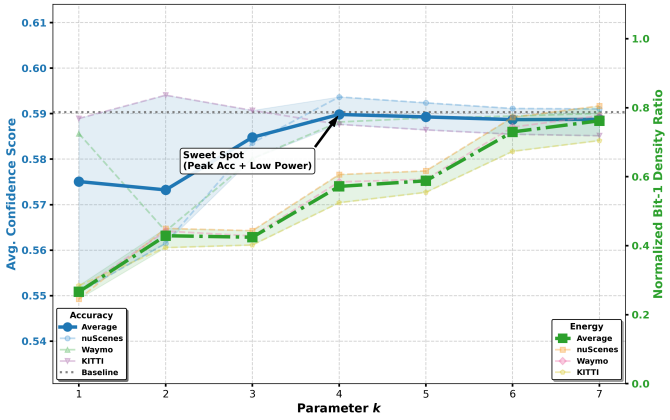

Fig. 5: **Design Space Exploration: Perception vs. Energy Trade-off varying bit-width k.** The plot sweeps the retained parameter k from 1 to 7. The blue line (Accuracy/Confidence) saturates around $k = 4$, while the green line (Energy Cost) continues to rise linearly. This identifies $k = 4$ as the algorithmic "Sweet Spot," providing maximum perceptual gain for the minimum necessary energy expenditure.

phases: a **Growth Phase** ($k = 1 \rightarrow 3$), where accuracy improves drastically as the top MSBs define object boundaries and shapes, and a **Saturation Phase** ($k \geq 4$), where accuracy plateaus, matching baseline performance. Increasing k beyond 4 offers negligible perceptual gain.

c) The "Sweet Spot" Decision: Based on this intersection, we identify $k = 4$ as the Pareto-optimal elbow point. It effectively filters out the high-frequency "noise" (lower 4 bits) that consumes energy but aids little in detection, while preserving the high-fidelity "signal" (upper 4 bits + inversion) required by the neural networks. Consequently, all subsequent comparisons (Pareto analysis in Fig. 1) use $k = 4$ as the default configuration for MotiMem.

C. Main Results

Table I presents the comprehensive comparison across 16 models and three datasets. We analyze the results focusing on the critical correlation between bit-statistics and hardware energy.

TABLE II: Relative mAP retention (%) across 16 detection models. Results averaged over 3 datasets. MotiMem achieves highest retention on all models.

Model	MotiMem	Glo. k=4	Uni. 4b	JPEG	WebP
YOLOv5n	**91.9**	90.4	90.5	91.7	89.8
YOLOv8n	**92.0**	90.6	90.6	91.6	89.7
YOLOv8s	**93.0**	91.7	91.7	92.7	90.7
YOLOv8m	**93.5**	91.9	91.9	91.8	90.0
YOLOv8l	**93.5**	92.1	92.2	91.3	90.1
YOLOv8x	**93.7**	92.3	92.3	91.3	89.9
YOLOv9c	**93.7**	92.2	92.2	91.3	90.3
YOLOv10n	**92.6**	91.2	91.2	92.8	90.8
YOLO11n	**92.3**	90.8	90.8	91.9	89.7
YOLO11s	**92.6**	90.8	90.8	90.9	89.2
YOLO11m	**93.7**	91.8	91.9	91.5	90.4
YOLO11l	**94.0**	92.4	92.4	91.8	90.6
YOLO11x	**94.2**	92.7	92.6	91.6	90.6
YOLO26m	**92.8**	90.8	90.9	90.5	88.9
RT-DETR-l	**89.2**	87.7	87.7	87.3	84.7
RT-DETR-x	**89.6**	88.0	88.0	87.8	85.2
Average	**92.6**	91.1	91.1	91.1	89.4

Glo.: Global, Uni. 4b: Uniform 4-bit, JPEG/WebP: Q50.

a) Energy-Accuracy Trade-off (The Physics of Saving): MotiMem establishes the optimal Pareto operating point. As detailed in Table I, our method reduces the **Normalized Bit-1 Density** to **0.57**. Since the dynamic power in memory buses is governed by the switching activity factor (α), and bit-1 density serves as a linear proxy for α, this density reduction **directly translates to a 43% saving in dynamic energy consumption**. In stark contrast, standard codecs fail to exploit this physical correlation: JPEG Q50 consumes significantly more energy (Density 0.93) for comparable accuracy, while JPEG Q100 even *increases* energy consumption (Density 1.12) due to high-entropy coding overhead, offering negative energy benefits despite volume compression.

b) Impact of Semantic Awareness (Ablations): The comparison with ablations further validates our design. Although the *Uniform 4-bit* variant achieves the lowest density (0.47), its lack of adaptive inversion leads to poor signal fidelity (PSNR 29.5 dB). Crucially, the semantically blind *Global k=4* baseline drops detection accuracy to 0.91 mAP. MotiMem recovers this loss (+0.02 mAP) by protecting the RoI with high-fidelity bits. Notably, MotiMem exhibits a lower PSNR (30.6 dB) than WebP Q50 (37.4 dB) yet achieves higher detection accuracy (0.93 vs. 0.89 mAP), proving that **optimizing the bit-density of semantic regions is more effective for energy-constrained**

Fig. 6: Qualitative comparison of detection results across three autonomous driving datasets (nuScenes, Waymo, KITTI) under different encoding methods. All images are processed by YOLO26m detector. MotiMem preserves detection accuracy comparable to the Original while significantly reducing energy consumption (43% reduction). JPEG Q10 exhibits severe compression artifacts leading to missed detections. Standard codecs (JPEG Q50, WebP Q50) maintain visual quality but offer no energy benefit.

perception than maximizing global pixel fidelity (PSNR).

c) Robustness Across Models and Datasets: MotiMem demonstrates consistent generalization. Across datasets, it maintains high accuracy on KITTI (0.95), Waymo (0.92), and nuScenes (0.91). Regarding model sensitivity, larger backbones exhibit higher resilience to our density-optimized encoding: **YOLO11x** achieves an impressive **94.2%** mAP retention. Even for sensitivity-prone Transformer models (RT-DETR), MotiMem maintains ≈89.6% retention, consistently outperforming standard codecs which suffer from block-artifact induced degradation.

D. Discussion and Limitations

The disconnect between PSNR and mAP in Table I (MotiMem: 30.6 dB/0.93 mAP vs. WebP: 37.4 dB/0.89 mAP) confirms a fundamental insight: for neural perception, background fidelity is often energy waste. By strictly prioritizing Motion-Guided RoIs, we demonstrate that semantic-aware bit allocation lowers the energy barrier for edge devices more effectively than statistical compression. However, we acknowledge two limitations: (1) Our 43% energy savings are based on analytical bit-density proxies ($P \propto \alpha C V^2 f$) and require future validation on physical silicon; (2) The reliance on $T-1$ feedback prevents optimization during "Cold Start" or abrupt object entries. The inflation margin δ is set to cover typical inter-frame motion and localization errors, providing tolerance for newly appearing objects near existing detections, but a global-precision fallback remains necessary for entirely novel scene regions.

V. Conclusion

We proposed **MotiMem**, a hardware-software co-designed interface that decouples semantic fidelity from background noise. By leveraging **Kinematic Priors** and **Hybrid Sparsity-Aware Coding**, MotiMem reduces the memory-interface dynamic energy proxy by $\approx 43\%$ while retaining $\approx 93\%$ of the detection mAP ($k = 4$), significantly outperforming standard codecs. Future work will focus on implementing the MotiMem logic on an **FPGA prototype** to validate physical power savings and extending the framework to **3D LiDAR point cloud perception**, where data sparsity can be further exploited for system-level efficiency.

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[2] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[5] G. Baris, B. Li, P. H. Chan, and C. A. Avizzano, "Automotive DNN based object detection in the presence of lens obstruction and video compression," *IEEE Access*, vol. 13, pp. 45 891–45 905, 2025.

[6] S. Yao, R. Guan, X. Huang, Z. Li, X. Sha *et al.*, "Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 11, pp. 4163–4180, 2023.

[7] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1697–1716, 2019.

[8] P. H. E. Becker, J. M. Arnau, and A. González, "Demystifying power and performance bottlenecks in autonomous driving systems," in *2020 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 2020, pp. 145–156.

[9] G. Kestor, R. Gioiosa, D. J. Kerbyson, and A. Hoisie, "Quantifying the energy cost of data movement in scientific applications," in *2013 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 2013, pp. 56–65.

[10] D. Pandiyan and C. J. Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," in *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2014, pp. 171–180.

[11] A. Saulsbury, F. Pong, and A. Nowatzyk, "Missing the memory wall: The case for processor/memory integration," *ACM SIGARCH Computer Architecture News*, vol. 24, no. 2, pp. 90–99, 1996.

[12] C. Zhang, H. Sun, S. Li, Y. Wang, and J. Lin, "A survey of memory-centric energy efficient computer architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 9, pp. 3643–3656, 2023.

[13] L. Guo, D. Zhou, J. Zhou, and S. Kimura, "Embedded frame compression for energy-efficient computer vision systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 2056–2069, 2018.

[14] Y. Wang, P. H. Chan, and V. Donzella, "Semantic-aware video compression for automotive cameras," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3456–3469, 2023.

[15] J. Löhdefink, A. Bär, N. M. Schmidt, F. Hüger *et al.*, "Focussing learned image compression to semantic classes for V2X applications," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[16] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 1, pp. 49–58, 2002.

[17] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A coding framework for low-power address and data busses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 2, pp. 212–221, 2002.

[18] P. Petrov and A. Orailoglu, "Low-power instruction bus encoding for embedded processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 8, pp. 812–826, 2004.

[19] H. Lekatsas, J. Henkel, and W. Wolf, "Approximate arithmetic coding for bus transition reduction in low power designs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 4, pp. 461–472, 2005.

[20] P. R. Panda and N. D. Dutt, "Low-power memory mapping through reducing address bus activity," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 3, pp. 309–320, 2002.

[21] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems," in *Proceedings of the 7th Great Lakes Symposium on VLSI*. IEEE, 1997, pp. 77–82.

[22] M. H. Faykus, B. Selee, J. C. Calhoun *et al.*, "Lossy compression to reduce latency of local image transfer for autonomous off-road perception systems," in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2022, pp. 217–224.

[23] P. Pawłowski and K. Piniarski, "Lossy compression of video sequences of automotive high-dynamic range image sensors for advanced driver-assistance systems and autonomous vehicles," *Electronics*, vol. 13, no. 18, p. 3651, 2024.

[24] F. Ponzina, S. Machetti, M. Rios, B. W. Denkinger *et al.*, "A hardware/software co-design vision for deep learning at the edge," *IEEE Micro*, vol. 42, no. 5, pp. 24–33, 2022.

[25] I. Karageorgos, K. Sriram, J. Veselý, M. Wu *et al.*, "Hardware-software co-design for brain-computer interfaces," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2020, pp. 999–1012.

[26] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, p. 660, 2021.

[27] P. Wu, S. Chen, and D. N. Metaxas, "Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 385–11 395.

[28] Y. Xie, H. Chen, G. P. Meyer, Y. J. Lee *et al.*, "Cohere3D: Exploiting temporal coherence for unsupervised representation learning of vision-based autonomous driving," *IEEE Transactions on Robotics*, vol. 41, pp. xxxx–xxxx, 2025, to appear.

[29] C. Doukas and I. Maglogiannis, "Region of interest coding techniques for medical image compression," *IEEE Engineering in Medicine and Biology Magazine*, vol. 26, no. 5, pp. 29–35, 2007.

[30] J. Liu, B. Zhang, and X. Cao, "ROI-aware dynamic network quantization for neural video compression," in *International Conference on Pattern Recognition (ICPR)*. Springer, 2024, pp. 1–8.

[31] Y. Song and E. Ipek, "More is less: Improving the energy efficiency of data movement via opportunistic use of sparse codes," in *Proceedings of the 48th International Symposium on Microarchitecture (MICRO)*. ACM, 2015, pp. 1–13.

[32] Y. Zhang, C. Yuan, L. Wang *et al.*, "The structure-preserving spectral graph neural network for dual kinase inhibitors and synergy scoring in gastric cancer," *npj Digital Medicine*, 2025.

[33] Z. Liu, C. Yuan, H. Liu *et al.*, "Mstdp: A multi-scale temporal deep learning framework for just-in-time software defect prediction with cross-attention fusion," *Journal of King Saud University - Computer and Information Sciences*, 2026.

[34] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 576–584.

[35] Y. Hao, H. Pei, Y. Lyu, Z. Yuan, J.-R. Rizzo, Y. Wang, and Y. Fang, "Understanding the impact of image quality and distance of objects to object detection performance," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11 436–11 442.

[36] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 8th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2016, pp. 1–6.

[37] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[38] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[39] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "Haq: Hardware-aware automated quantization with mixed precision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8612–8620.

[40] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv preprint arXiv:2103.13630*, 2021.

[41] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations (ICLR)*, 2018.

[42] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[43] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[44] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[45] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An end-to-end deep video compression framework," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[46] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[47] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[48] "ISO/IEC 10918-1: Information technology — digital compression and coding of continuous-tone still images: Requirements and guidelines," International Organization for Standardization, 1994.

[49] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[50] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, 1992.

[51] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 1, pp. 49–58, 1995.

[52] M. Liu, H. Que, X. Yang, K. Zhang, Q. Yu, L. Yan, T. Wang, Y. Jin, and N. Zhou, "A selective bit dropping and encoding co-strategy in image processing for low-power design in dram and sram," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 48–57, 2023.

[53] M. Liu, S. Feng, W. Shan, H. Que, J. Wang, and X. Yang, "A new low-power circuit design optimization for image processing." *Electronics (2079-9292)*, vol. 14, no. 2, 2025.

[54] H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 booth multipliers for low-power and high-performance operation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2638–2644, 2015.

[55] W. Liu, Q. Liao, F. Qiao, W. Xia, C. Wang, and F. Lombardi, "Approximate designs for fast fourier transform (fft) with application to speech recognition," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 12, pp. 4727–4739, 2019.