

Quanta Neural Networks: From Photons to Perception

Varun Sundar^{†*}
vsundar4@wisc.edu

Tianyi Zhang[‡]
tz12@rice.edu

Sacha Jungerman[†]
sjungerman@wisc.edu

Mohit Gupta[†]
mohitg@cs.wisc.edu

[†]University of Wisconsin-Madison [‡]Rice University

wisionlab.com/project/quanta-neural-networks/

Abstract

Quanta image sensors record individual photons, enabling capabilities like imaging in near-complete darkness and ultra-high-speed videography. Yet, most research on quanta sensors is limited to recovering image intensities. Can we go beyond just imaging, and develop algorithms that can extract high-level scene information from quanta sensors? This could unlock new possibilities in vision systems, offering reliable operation in extreme conditions. The challenge: raw photon streams captured by quanta sensors have fundamentally different characteristics than conventional images, making them incompatible with vision models. One approach is to first transform raw photon streams to conventional-like images, but this is prohibitively expensive in terms of compute, memory, and latency.

We propose quanta neural networks (QNNs) that directly produce downstream task objectives from raw photon streams. Our core proposal is a trainable QNN layer that can seamlessly integrate with existing image- and video-based neural networks, producing quanta counterparts. By avoiding image reconstruction and allocating computational resources on a scene-adaptive basis, QNNs achieve 1–2 orders of magnitude improvements across all efficiency metrics (compute, latency, readout bandwidth) as compared to reconstruction-based quanta vision, while maintaining high task accuracy across a wide gamut of challenging scenarios including low light and rapid motion.

1. Vision Without the Image

Much of computer vision today begins with an image—the de-facto representation of light, driven by the proliferation of digital cameras. A nascent class of sensors, called *quanta image sensors* [5, 18, 19, 60] is challenging this status quo by providing the ability to sense light at its fundamental level of discretization: individual photons. Sensing raw photons facilitates imaging in extreme scenarios spanning very low light [4, 52] to extremely fast motion [6, 73]. Quanta sensor hardware has evolved tremendously in the past 5 years, going from single-pixel prototypes to megapixel-resolution arrays sensors [19, 21, 47, 53, 54, 69],

making computer vision *from photons* an increasing possibility. However, do we have the algorithms and models for performing high-quality computer vision and perception from photons?

Quanta cameras capture photon detections as a series of binary-valued frames. As seen in Fig. 1 (*first row*), a single quanta frame is quite different from a conventional image; it is heavily quantized and very noisy, making it incompatible with off-the-shelf computer vision algorithms. Fortunately, quanta cameras can run at very high speeds (~100 kHz), so we can aggregate multiple frames over time by aligning them and reconstruct images that are interpretable by existing vision models [6, 8, 51, 61]. Unfortunately, intensity reconstruction from quanta frames can be an expensive undertaking that can severely strain the compute, memory, and latency of an edge-computing platform by around 2–3 orders of magnitude, precluding the adoption of quanta sensors in resource-constrained applications.

In this work, we design quanta neural networks (QNNs) that *directly* transform photon detections to downstream task objectives, eschewing intermediate image reconstruction. Our core contribution is a learnable temporal layer, or QNN layer, that can be incorporated throughout the design of many (feedforward) deep neural networks, yielding their photon-level equivalents (illustrated in Fig. 1 (*second row*)). QNNs can directly benefit from the state-of-the-art techniques developed by the broader computer vision community today, and possibly even in the future. As case studies, we provide quanta neural network designs for one image-based and two video-based neural networks.

Central to our design philosophy is computational efficiency. QNN layers produce outputs with reduced latency and constant memory footprint, even when operating on long temporal durations, by employing streaming and recursive computations. A second property of QNNs is their scene-adaptive resource allocation: expending more computations when rapid motion is involved, and suppressing computations otherwise. We achieve this using two complementary inference modes. *Eventful computations* reduces floating-point operations by an order of magnitude by updating QNN feature maps incrementally, exploiting temporal redundancy at the neuron granularity. *Change-driven inference* automatically runs the entire network at slow or brisk rates as per the scene dynamics, allowing QNN output rates to vary across a spectrum of 20 Hz to 100 kHz.

*This research was supported in part by NSF CAREER award 1943149, and Wisconsin Alumni Research Foundation via a Research Forward Initiative.

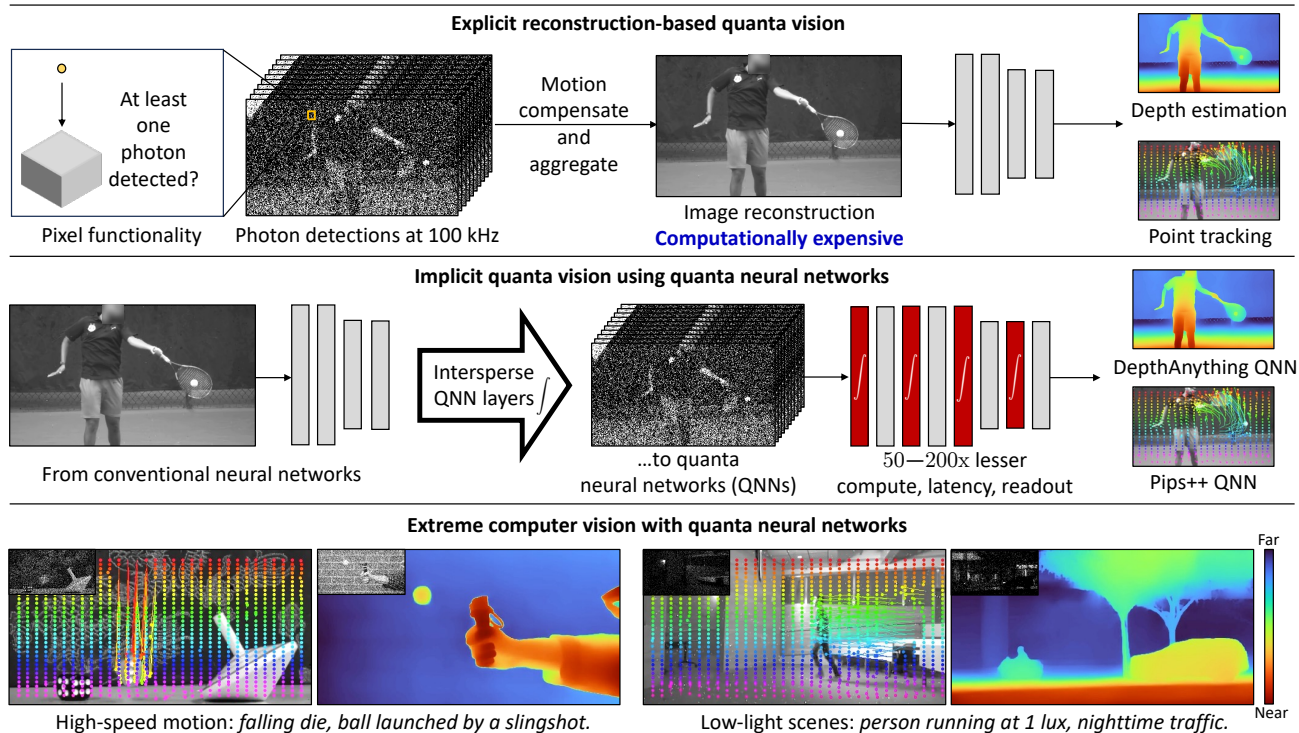


Figure 1. **Quanta neural networks.** (*Top row*) A quanta image sensor captures individual photon detections at very high speeds. However, a single quanta frame contains too little information to directly run perception on. One approach is to first reconstruct images, but this approach is computationally demanding. (*Second row*) This work converts image-based (e.g., DepthAnything-v2 [76]) and video-based (e.g., Pips++ [83]) neural networks to quanta neural networks (QNNs) by suitably interspersing QNN layers that perform temporal aggregation. QNNs directly transform photon detections to downstream tasks without reconstructing images, which coupled with our proposed scene-adaptive inference modes, improves efficiency (across compute, latency, and readout) by 2 orders of magnitude. (*Third row*) As case studies, we demonstrate the capabilities of point-tracking and depth-estimation QNNs on scenes involving fast motion and low light. Insets show a single quanta frame from each sequence.

Readout considerations: sensor-proximal QNN layer. Sensing raw photons, in addition to its core computational difficulty, also introduces nearly $1000\times$ more data compared to a conventional camera. A promising approach to tackle this data deluge is to move compute closer to the sensor, via near-sensor processing. We devise a lightweight instantiation of our general QNN layer that uses operations comparable to near-sensor algorithms [66, 67]. This sensor-proximal QNN layer can compress photon detections by 2 orders of magnitude.

Can quanta cameras and QNNs run on my phone? Not yet. All in all, QNNs provide $50\text{--}200\times$ reduction in compute, latency, and readout over reconstruction-based quanta vision [8, 51, 67]. Using a creative analogy, this reduction is the difference between needing a beefy NVIDIA A100 GPU versus a relatively inexpensive Jetson Nano processor. However, there is another order of magnitude or two reductions left for QNNs to achieve edge-device photon perception. Part of this reduction could come from increased support for sparse operations, that our inference modes utilize on commodity processors.

2. Related Work

Imaging capabilities of quanta sensors. By reconstructing images from photon detections, quanta sensors have demonstrated

low-light [4, 50, 52], high-dynamic range [32, 33], and high-speed imaging [6, 51]: thereby serving as “all-scenario” imaging devices. Such reconstruction techniques may involve motion compensating and merging quanta frames [34, 35, 51], deblurring using motion-adaptive integration windows [38, 39, 61], or more recently, learned neural networks [8].

Direct perception with quanta sensors. Prior works that look at reconstruction-free perception with quanta sensors are mainly designed for Jot-based quanta sensors and process a few (1–8) quanta frames [20, 40]. A few works consider high-speed quanta frames like this work for direct perception, but assume no motion during capture [22], or produce feature-bank responses that are compatible with (handcrafted) phase-based computer vision algorithms [26]. In contrast, our work provides a broadly applicable recipe—based on a temporal integration or QNN layer that aggregates high-speed quanta frames over long temporal histories (1000s of frames)—to transform many image- and video-based neural networks to quanta-sensor counterparts.

Practical imaging with quanta-image sensors. Quanta-sensor readout can be significantly reduced by compressing photon detections on near-sensor processors before data transmission, e.g., using event-camera-inspired selective readout [66, 67] or sketching using exponential smoothing [80]. While these works

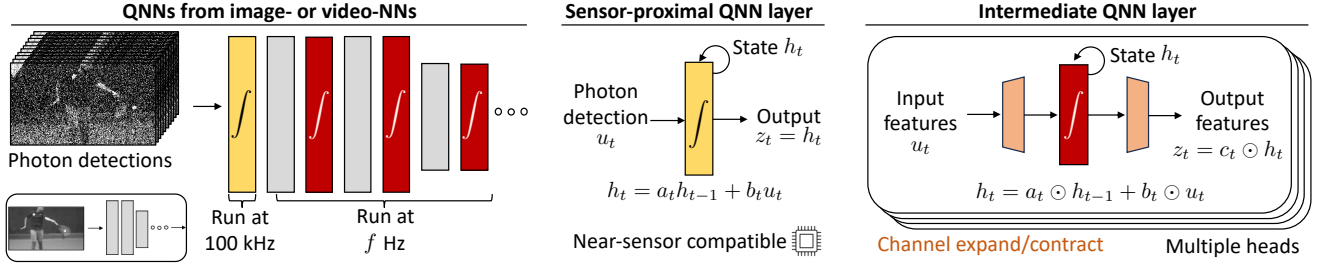


Figure 2. **Transforming image-based neural networks to quanta neural networks.** (left) We insert QNN layers (\int symbol) between the layers of an image-based (or video-based) neural network, which are depicted as red blocks. QNN layers use scalar state-space equations to temporally aggregate feature maps. These layers may operate at a frequency (e.g., f Hz) that differs from the quanta camera’s framerate and can be altered at inference time, owing to the QNN layer’s continuous-time formulation. (center) The first QNN layer used at the sensing stage is a simple instantiation of the more general layer, consisting of a single scalar recurrence that operates on photon detections (quanta frames). With particular choices for a_t and b_t , the sensing QNN layer adaptively accumulates photons based on the level of scene motion. (right) Intermediate QNN layers employ multiple scalar equations, depending on the number of “heads” used, that operate on feature map tensors (spatial and channel dimensions).

improve the bandwidth efficiency of quanta sensors, their downstream processing costs remain high. In comparison, our proposed QNNs reduce inference costs by more than an order of magnitude while featuring similar modest readout numbers.

3. Building Quanta Neural Networks

At the heart of a quanta neural network is a temporal integration layer, or QNN layer, that can be flexibly introduced in the design of a deep learning architecture—enabling it to operate *directly* on photon detections. Fig. 2 provides a high-level illustration of QNN design. The first or *sensor-proximal* QNN layer aggregates photons, while subsequent QNN layers temporally aggregate feature maps. In fact, the first layer is a lightweight special case of the more general QNN layer. All QNN layers provide streaming feature representations that can be queried at any time and are amenable to the efficient inference modes that we present in Sec. 4. We begin our exposition with the sensor-proximal layer, whose temporal modeling can be interpreted as a motion-aware approach to integrating photons.

3.1. Background: Photon Detection Model

A quanta image sensor operates as a high-speed photon detector, capturing 1-bit frames at high speeds, at around 100 kHz or higher. Let $N_t(\mathbf{p})$ be the average number of photons incident on a quanta sensor at pixel location \mathbf{p} and frame index t . The sensor outputs a binary value $B_t(\mathbf{p})$, corresponding to whether at least one photon was detected at the pixel location, which can be modeled as a Bernoulli random variable [75]:

$$\Pr \{B_t(\mathbf{p}) = 1\} = 1 - e^{-N_t(\mathbf{p})}. \quad (1)$$

The techniques in this work can carry over, with suitable modifications, to quanta sensors that output few-bit values [47–49].

3.2. A Sensor-Proximal Layer

A key function of the first layer is to aggregate photons over time—which sounds simple at first glance, but is a challenging problem. If the layer integrates too little, it leads to severe noise;

whereas too much integration causes severe blur (Fig. 3 (top row)) resulting in loss of visual information. It is therefore critical to integrate photons *adaptively*, as a function of the level of motion and scene content. Given these, we propose an adaptive integrator $\mathcal{I}_{\text{adapt}}$ that performs time-varying exponential smoothing [15, 68] at each pixel location:

$$\mathcal{I}_{\text{adapt},t}(\mathbf{p}) = \omega_t(\mathbf{p}) \mathcal{I}_{\text{adapt},t-1}(\mathbf{p}) + (1 - \omega_t(\mathbf{p})) B_t(\mathbf{p}), \quad (2)$$

where $0 \leq \omega_t \leq 1$ determines how $\mathcal{I}_{\text{adapt}}$ weighs recent history. For example, setting $\omega_t(\mathbf{p})$ to be a constant value corresponds to regular exponential smoothing which leads to the noise-blur tradeoff described earlier. Our goal is to design $\omega_t(\mathbf{p})$ that ideally overcomes this tradeoff. There is a causality dilemma to this problem: to derive $\omega_t(\mathbf{p})$, we must robustly measure flux variations, but these measurements in turn depend on $\omega_t(\mathbf{p})$.

To address this challenge, we turn to an online Bayesian algorithm [1] that estimates the time since the last abrupt change, or the *run length* $r_t(\mathbf{p})$, of a Bernoulli time series. Specifically, at each pixel location \mathbf{p} , we maintain S (typically 5–10) forecasters $\{\nu_s(\mathbf{p})\}_{s=1}^S$ that are initialized at times $\{t_s\}$, and whose values denote the probability of the run length at time t being $t - t_s$. Using these forecasters, we estimate run length

$$r_t(\mathbf{p}) = \left(\sum_{s=1}^S \nu_s(\mathbf{p})(t - t_s) \right) / \sum_{s=1}^S \nu_s(\mathbf{p}). \quad (3)$$

Each forecaster $\nu_s(\mathbf{p})$ is associated with a uniform Beta prior, i.e., $\text{Beta}\{\alpha_s(\mathbf{p}) = 1, \beta_s(\mathbf{p}) = 1\}$. As more photon detections are processed, the associated prior is updated as $\alpha_s(\mathbf{p}) \leftarrow \alpha_s(\mathbf{p}) + B_t(\mathbf{p})$ and $\beta_s(\mathbf{p}) \leftarrow \beta_s(\mathbf{p}) + 1 - B_t(\mathbf{p})$. In turn, the forecaster is updated as:

$$\nu_s \leftarrow (1 - \gamma) \nu_s \left(\frac{\alpha_s}{\alpha_s + \beta_s} B_t + \frac{\beta_s}{\alpha_s + \beta_s} (1 - B_t) \right), \quad (4)$$

where we drop the pixel index \mathbf{p} for brevity, and $0 \leq \gamma \leq 1$ controls the responsivity of run-length modeling (higher biases towards shorter run lengths). We provide details on forecaster initialization and updates in the supplementary. From the

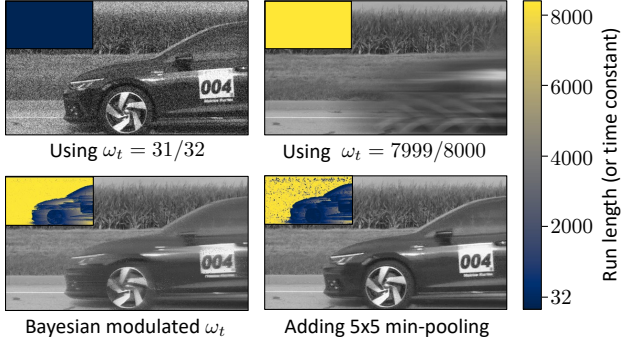


Figure 3. **Adaptive aggregation modulates per-pixel exposures.** (*top row*) Without adaptive integration (or setting ω_t to a constant), there is a strong tradeoff between preserving motion and reducing photon noise, (*bottom row*) $\mathcal{I}_{\text{adapt}}$ picks per-pixel exposures according to the level of scene motion. To reduce blur, we resort to min-pooling run-length estimates using a kernel size of 5×5 and stride 1. Insets plot the run lengths, *i.e.*, the exponential-smoothing time constant, $1/\log(1/\omega_t)$.

run-length estimate, we set the smoothing parameter $\omega_t(\mathbf{p}) = e^{-1/r_t(\mathbf{p})}$. In words, we set the time constant of exponential smoothing at each instant as per the estimated run length.

Fig. 3 compares adaptive to fixed aggregations, that is, setting ω_t dynamically or to a constant value. In practice, for more responsive motion modulation, we find it beneficial to “min pool” run-length estimates using a kernel size of 5×5 or 7×7 before setting the smoothing parameter, *i.e.*, use the minimum run length across a neighborhood (Fig. 3 (*bottom row*)). Our adaptive aggregator-based QNN layer preserves scene motion while substantially reducing photon noise, which provides more information for downstream processing, and translates to computational and communication efficiency (Sec. 4).

Adaptive exponential smoothing is a selective state space model. It turns out that there is an intriguing connection between our adaptive aggregator ($\mathcal{I}_{\text{adapt}}$) and state space models that are the workhorse of several recent time-series models such as Mamba [10, 23]. Among these, a scalar state space model [10] maps an input time series $u_t \in \mathbb{R}$ to state $h_t \in \mathbb{R}$:

$$h_t = a_t h_{t-\Delta t} + b_t u_t, \quad (5)$$

where Δt is the time since the state-space model was last run, and a_t and b_t denote time-varying state-space scalars that are parameterized as

$$a_t = e^{-\Delta t f(u_t)}, \quad b_t = (1 - a_t)g(u_t), \quad (6)$$

where $f : \mathbb{R} \rightarrow \mathbb{R}^+$ and $g : \mathbb{R} \rightarrow \mathbb{R}$ are learnable functions. This particular form of Eq. (6) results from the zero-order hold discretization of an underlying continuous-time differential equation (which we show in the supplementary material).

The proposed $\mathcal{I}_{\text{adapt}}$ is a selective state space model that *operates on Bernoulli time series* $B_t(\mathbf{p})$ using the parameters $a_t = \omega_t(\mathbf{p}) = e^{-1/r_t(\mathbf{p})}$ and $b_t = 1 - \omega_t(\mathbf{p}) = 1 - e^{-1/r_t(\mathbf{p})}$. Here, $r_t(\mathbf{p})$ is the estimated runlength, a function of $\{B_\tau(\mathbf{p})\}_{\tau=1}^t$.

Motivated by this connection, in the next section, we will apply state-space models as a learnable temporal layer, *i.e.*, the QNN layer that operates on neural-network feature maps.

3.3. Intermediate QNN Layers

While our first QNN layer operates on quanta frames, intermediate QNN layers operate on time-varying feature maps $u_t \in \mathbb{R}^{(H,W,C)}$, where H and W are the spatial dimensions, and C is the channel dimension. Like Mamba-2 [10], we associate each element u_t with a recursively-updated state $h_t \in \mathbb{R}^{(H,W,C)}$:

$$h_t = a_t \odot h_{t-\Delta t} + b_t \odot u_t. \quad (7)$$

Here, a_t and b_t are functions of the input u_t as specified by (a broadcasted version of) Eq. (6), Δt is the number of quanta frames since the previous state update, and \odot denotes the Hadamard (element-wise) product. We specify f and g in Eq. (6) using learnable linear parameterizations, *i.e.*, $f(u_t) = \text{ReLU}(w_f u_t + b_f)$ and $g(u_t) = w_g u_t + b_g$, where w_f, w_g, b_f and b_g are learnable scalars, and addition (“+”) is broadcasted. Finally, we output

$$z_t = c_t \odot h_t, \quad \text{where } c_t = w_c \odot u_t + b_c, \quad (8)$$

and w_c and b_c are learnable scalars. Overall, the QNN layer operates in a broadcasted manner, *i.e.*, each element of the input u_t is point-wise transformed to an element of the output z_t .

To increase layer capacity, we channel expand u_t before applying the state-space model, producing $\tilde{u}_t \in \mathbb{R}^{(H,W,CN)}$, where N is the channel expansion factor. We use a linear projection layer operating in the channel dimension to map C channels to CN channels, akin to a 1×1 convolutional layer. Further, we stack D copies, or “heads”, of these state-space models—like multi-head attention—and operate on a channel dimensionality of CND ; channel stacking and multiple heads are depicted in Fig. 2 (*right*). We treat D (nominally set to 4) as a hyperparameter that can be used to alter the deep network’s model capacity.

Can we design QNN layers based on recurrent neural networks (RNNs)? Although a valid choice, RNNs lack key properties that we shall exploit to facilitate efficient inference. First, the state-space layer involves point-wise operations and is considerably cheaper to compute than other layers in a deep network, *e.g.*, convolutions or spatial transformer blocks. Thus, for boosting the inference efficiency of QNNs, we can focus on reducing the inference cost of these convolutional and transformer layers. This observation is also true for certain RNN variants where state updates involve diagonal matrix multiplications [70], but not for vanilla RNNs featuring dense matrix multiplications. Second, the state-space model has a continuous-time interpretation, allowing us to train QNN layers at a particular frame rate, but infer at a different cadence or at irregularly spaced time instants. We also inherit some auxiliary benefits from using (scalar) state-space models [10]—including GPU-parallelizable algorithms for training, unlike RNNs that suffer from slow sequential training.

4. QNNs On-a-Budget

Sec. 3 explains the basic building blocks of QNNs, and how they aggregate information temporally for robust perception. However, consider a quanta sensor running at 100 kHz. Incurring even a cost of 1 ms to process a single frame, an acceptable latency for many edge-deployed neural networks, would imply taking 100s to process one second of quanta-sensor acquisition. Thus, we simply cannot run QNNs directly on quanta frames as is. How can we achieve the benefits of QNNs while staying within practical resource and latency envelopes? Our key observation is that while the sensor-proximal QNN layer accepts photon streams at extremely high speeds, later QNN layers transform those into feature maps and eventual task objectives, which typically do not vary at such high speeds—resulting in an inherent temporal redundancy across the QNN layers.

One direct approach to exploit this temporal redundancy, which forms our base inference mode or “QNN dense”, is to subsample outputs of the sensor-proximal layer, *i.e.*, the initial layer consumes quanta frames at the sensor’s frame rate and the rest of the QNN stack *subsamples* its outputs. We achieve this by altering Δt used in the state-space equations Eq. (7). Owing to their continuous-time interpretation, QNNs can generalize to time gaps different from train-time settings.

However, this direct approach is scene-agnostic, which can be wasteful for slow-moving scenes and detrimental when there is fast motion. We design two complementary approaches that exploit temporal redundancy by allocating computational resources in a scene-adaptive manner. *Eventful computation* operates layers of a quanta neural network incrementally, on the changes to its input feature maps. Eventful computation also leads to parsimonious motion-dependent readout, which is reminiscent of event cameras [3, 45]. *Change-driven sampling* allows us to flexibly choose output times that can be irregularly spaced instants and driven by the underlying scene dynamics.

4.1. Eventful Computations and Readout

Many deep neural network layers combine a linear operator (denoted by \mathcal{L} , *e.g.*, fully connected layers, convolutions, matrix-vector products in self-attention) and a point-wise non-linear activation (*e.g.*, ReLU, sigmoid) that is more lightweight to compute. Linear operators can be updated incrementally, *i.e.*, $\mathcal{L}(x + \delta x) = \mathcal{L}(x) + \mathcal{L}(\delta x)$. Thus, if $\mathcal{L}(x)$ is cached (*e.g.*, the output at a previous instant), and running $\mathcal{L}(\delta x)$ is cheap, then computing $\mathcal{L}(x + \delta x)$ can be cheap. Non-linear activations continue to be run densely, by accumulating changes into values.

QNN layers employ lightweight element-wise computations, so we run their core recurrence (Eq. (7)) on dense values (instead of increments). We sandwich linear operations in QNNs—including operations that drive the selective parameters a_t , b_t and c_t (Eqs. (6) and (8))—between “gating” and “accumulate” layers [13, 86]. Concisely, gating selects significant changes in an input tensor using an event policy, *e.g.*, highest $K\%$ of values in the input tensor, which we term as a “top- $K\%$ ” pol-

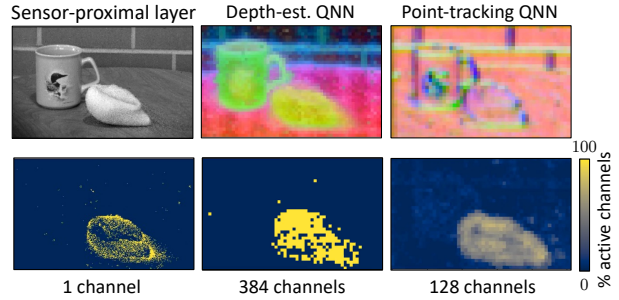


Figure 4. **Eventful computation** using a top-10% policy, visualized for the sensor-proximal stage and intermediate feature maps. For transformer models (depth estimation), gating operates in the spatial dimensions. For convolutional models (point tracking), channel sparsity is also used. The event policy reduces FLOP counts by $9\times$ for both models. Features are visualized using their top-3 principal components.

icy. Accumulate layers sum up the result of linear operations on increments and present dense versions to subsequent layers. Please see the supplementary material for more detailed descriptions of gating and accumulation. Finally, since QNNs operate causally, we apply eventful computations after an initial number of quanta frames (~ 200 , or about 2 ms) have been processed, allowing QNN states to filter out photon noise.

Eventful computations can reduce FLOP counts by $5\text{--}10\times$, depending on the level of motion, without appreciably lowering performance. Fig. 4 shows eventful computations in transformer (depth-estimation) and convolutional (point-tracking) QNNs; architecture details are provided in Sec. 5.1. By altering the event policy (value of K), we can gracefully trade performance to reduce compute costs—we show this in Sec. 5.2.

Sparse readout. Applied to the sensor-proximal QNN layer, eventful encoding can help reduce readout costs. We transmit sparse matrices using compressed-sparse column/row formats (since $\mathcal{I}_{\text{adapt}} \in \mathbb{R}^{(H,W)}$) along with the time-stamp information (per encoded matrix). As we show in Sec. 5.2, eventful QNN readout leads to a superior rate-performance tradeoff than prevailing bandwidth-efficient quanta-imaging techniques [67].

4.2. Change-driven (Irregular) Sampling

Eventful computations are an effective tool to reduce floating-point operations but require specialized sparse-workload accelerators (or neuromorphic hardware) to fully realize wall-time speedups. We now propose a complementary *change-driven sampling* mode that offers substantial run-time advantages by adaptively determining output rates.

We let inference be driven by significant changes in the sensor-proximal stage, thereby automatically adjusting resource costs according to the level of motion. Specifically, we run the QNN stack (beyond the initial layer) whenever the fraction of significant changes in the adaptive integrator $\mathcal{I}_{\text{adapt}}(\mathbf{p})$,

$$\sum_{\mathbf{p}} \mathbb{I}\{|\mathcal{I}_{\text{adapt}}(\mathbf{p}) - \mathcal{I}_{\text{ref}}(\mathbf{p})| \geq \tau\} / \sum_{\mathbf{p}} 1, \quad (9)$$

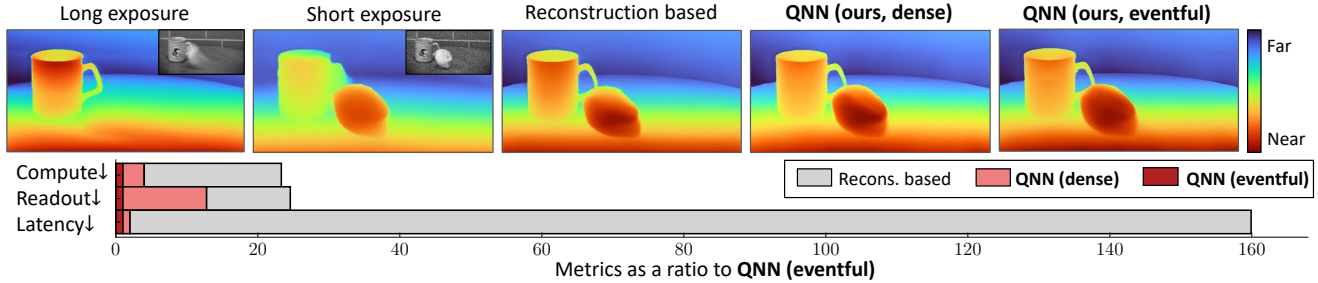


Figure 5. **Monocular-depth estimation of a stress-ball deforming at high speeds.** Long and short exposures (insets) are too blurred or noisy to accurately estimate relative depths. Explicit-image restoration, here using video denoising [41], followed by depth estimation succeeds, but with high compute (502 GFLOP/output), readout (9075 bits/pixel/sec or bps) and latency (6000 ms) costs. Our DepthAnything-v2 QNN with dense computations accurately estimates depth with 78 GFLOP/output and 38 ms latency; however, readout is unchanged. With eventful computations and readout, costs further reduce to 26 GFLOP/output and 770 bps. Efficiency metrics are presented as a ratio to the QNN eventful approach.

exceeds a critical value (*e.g.*, 5%). Here, \mathbb{I} is the indicator function, \mathcal{I}_{ref} is the previously-stored value of $\mathcal{I}_{\text{adapt}}$, and τ is a threshold (set between 0.1–0.2). Further, we set Δt to be the gap between the irregular inference times. Change-driven sampling can provide substantial speedups when a scene consists of a wide range of motion speeds: *e.g.*, when capturing high-speed dynamics, the phenomena of interest may only last a short duration. In Sec. 5.3, we demonstrate how change-driven inference adapts between a slow 20 Hz to an ultra-fast 24000 Hz without knowing any scene-specific information ahead of time.

5. Experimental Validation of QNNs

We demonstrate the inference capabilities of quanta neural networks on three computer vision tasks where we *lift* image- and video-neural networks to photon-level equivalents by introducing QNN layers. These tasks can be seen as example case studies; QNN layers are compatible with a broad swath of neural networks. To demonstrate the real-world applicability of the proposed approaches, we show results on real data acquired with a SwissSPAD2 [69] quanta sensor. The sensor consists of an array of 256×512 pixels which we operate at 96.8 kHz.

5.1. QNNs In-the-Wild

Using three example tasks, we showcase the versatility of QNNs in enabling computer vision across a spectrum of illumination and motion scenarios, spanning a dimly lit room to sports-photography speeds. These results involve inference across quanta-frame sequences, but for brevity, we show results at a single-frame index; please see the supplementary material for multi-frame outputs. When comparing to baseline methods, we assume evenly-sampled inference instants: we analyze the utility of change-driven irregular inference in Sec. 5.3.

Monocular depth estimation. We pick the DepthAnything-v2-small model [76] and insert QNN layers between its transformer blocks. To retain the original model’s strong generalization capabilities, we initialize with its weights and only finetune the QNN and the depth-prediction layers [58]. When finetuning, we assume $128\times$ subsampling at the sensor-proximal layer and use

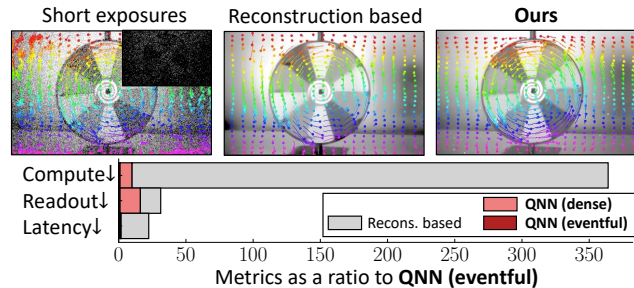


Figure 6. **Tracking points on a spinning prize wheel in low light.** The light level here is 3 lux (on the sensor side), or 0.07 PPP (quanta frame shown in the *top-left inset*). We use QUIVER [8] as our restoration-based point-tracking baseline. Our (eventful) QNN consumes $350\times$ fewer FLOPs compared to restoration-based computer vision.

a Blender-simulated dataset consisting of quanta frames with an average photons-per-pixel (PPP) in the range (0.05, 0.5), and paired depth maps. Further, we employ a combination of scale-and-shift-invariant and gradient-matching loss terms [44].

Fig. 5 shows a scene with non-rigid deformations of a stress ball which are challenging to capture without high-speed acquisition. The acquisition comprises 8192 quanta frames from which we output depth maps at 64 time instants. Running DepthAnything-v2 directly on summed photons results in either the dynamics being missed (strong blur due to long exposure) or erroneous depth (short exposure). We could run image reconstruction from quanta frames sequence before depth estimation, but it comes with extremely high computational costs. Moreover, since we have to wait for reconstruction to finish before running depth estimation, the latency, or time to the first output, is prohibitively high. In contrast, the proposed QNN approach requires $20\times$ fewer computations and $150\times$ lower latency, by running in a streaming (or recurrent) fashion, while achieving high accuracy depth estimation even in this challenging scene with high-speed non-rigid deformations. With eventful computations, *i.e.*, by updating feature maps incrementally, we can further reduce FLOP counts by $5\times$ and readout by $10\times$.

Multi-frame point tracking. We employ the Pips++ model [83] which tracks a set of query points throughout a video. We insert

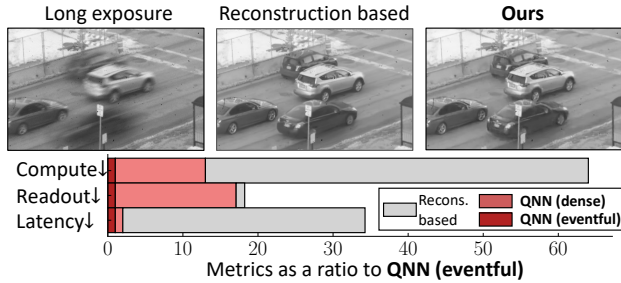


Figure 7. **Videography of a traffic scene.** We compare our method to a bandwidth-efficient videography method [67]. QNNs improve by as much as 50 \times on the compute, latency, and memory metrics, and offer matching bandwidth numbers when using their sparse readout mode.

QNN layers between the feature-extractor blocks of Pips++ and leave the track-refinement block as is. For a training dataset, we simulate quanta sequences of length 8192 with annotated tracks using the Kubric simulator [37]. During training, we set the subsampling factor to 256 \times , sample PPP values in the range (0.05, 0.5), and use the weighted L_1 loss objective [27].

We perform point tracking on a spinning tabletop wheel in low light (3 lux). Using the SwissSPAD2, we capture 4096 quanta frames, and we track points across 64-evenly spaced instants. We consider QUIVER [8] for our reconstruction-based point tracking baseline. As seen in Fig. 6, running Pips++ on noisy short exposures leads to erroneous tracks. Reconstruction-based point tracking produces reliable tracks but involves high inference costs. Since QUIVER is a recurrent network, we estimate its latency as the processing time for a single sum image (of 80 quanta frames), which we note is lower than the latency of our previous video-denoising baseline. Our QNN approach produces high-accuracy streamlined tracks while incurring 350 \times lower FLOP counts and 30 \times lesser data readout.

Intensity restoration. While not our main objective, we validate the proposed QNN approach on intensity restoration as an example end task. We compare against a bandwidth-efficient quanta videography technique [67], by adopting its restoration network; we replace its time-domain transformer layers with QNN layers (of similar number of heads and head dimensionality), and 3D convolutions with 2D convolutions. During training, we subsample the sensor-proximal QNN layer by 64 \times .

For a training dataset, we simulate quanta frames using the XVFI high-speed video dataset [63]. We logarithmically sample the average photons per pixel (PPP) in the range (0.05, 0.5) and train for 40 epochs using the L_1 loss objective.

Fig. 7 shows intensity reconstruction on a traffic sequence spanning 42 ms and at 1500 FPS. The long exposure depicts the range of motion. While the bandwidth-efficient reconstruction method offers modest readout, it involves large compute and latency costs. In comparison, QNNs run recurrently, resulting in 4 \times lower FLOP counts and 32 \times lower latency. Further, with eventful computations, we see an order of magnitude reduction in inference FLOP count and low readout costs.

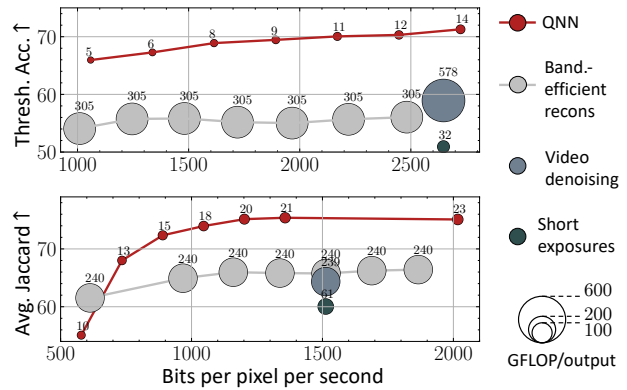


Figure 8. **Rate- and compute-performance tradeoff** as evaluated for the depth-estimation (*top row*) and point-tracking (*bottom row*) tasks. Our QNN approach offers better performance for both tasks (higher is better for both metrics) across a sweep of readout rates, while reducing floating-point operations by one to two orders of magnitude. Marker annotations here denote GigaFLOP count per output.

5.2. Compute- and Rate-Performance Tradeoff

In this section, we study the trade-off between the resource requirements of QNNs and their task performance on depth estimation and point tracking. We compare against two baselines that we use for reconstruction-based quanta vision: video denoising and bandwidth-efficient videography. The latter can vary its readout amount by altering a parameter.

We measure task performance on the test splits of the datasets simulated for the depth estimation and point-tracking tasks. Each data point features 4096 quanta frames simulated at 64 kHz with an average of 0.3 PPP. For depth estimation, we use a sensor-proximal subsampling of 64 \times , and measure task performance using threshold accuracy (percentage of points where the estimated and ground truth depths differ by not more than 25%). For point tracking, we subsample by 256 \times , and calculate the average Jaccard index when tracking 512 query points per sequence with the most motion; this measures the percentage of predicted tracks that lie within a threshold distance of the ground truth. Following Harley et al. [27], for (quanta) frames of spatial resolution 256 \times 256, we average the Jaccard index over the thresholds {1, 2, 4, 8, 16}.

As Fig. 8 shows, QNNs offer more than one order of magnitude FLOP reduction with the option to gracefully trade-off performance to reduce computational costs—a flexibility that the baselines do not possess. Our method simultaneously provides a rate-performance tradeoff superior to current bandwidth-efficient restoration methods; this is particularly pronounced for depth estimation, partly because QNNs model temporal dynamics throughout their architecture, while the baselines estimate depths on a per-frame basis.

We also present preliminary *wall-time savings* for eventful inference, when running on CPU, in the supplement. Realizing GPU speedups requires carefully minimizing sparse-computation overheads and is an important next step.

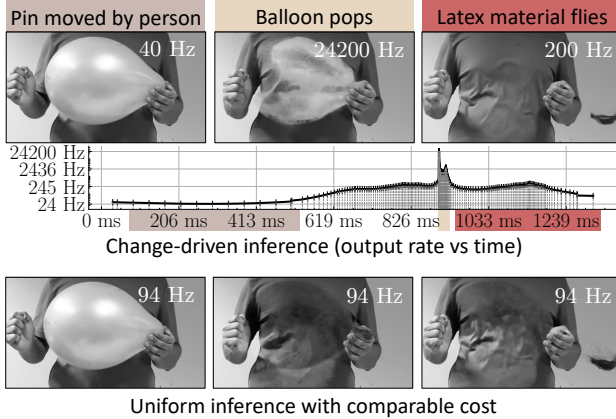


Figure 9. **Change-driven inference instants.** (*Top row*) We run inference based on significant changes in the sensor-proximal stage ($\mathcal{T}_{\text{adapt}}$). Dotted-gray bars indicate the quanta frame indices when inference occurs. We run inference more rapidly as the balloon pops than before and after, yielding a $150\times$ run-time speedup over evenly-spaced inference that can capture the same dynamics. (*Bottom row*) We further note quality improvements over uniform inference of comparable cost.

5.3. Change-driven Sampling

High-speed dynamic phenomena tend to be ephemeral, making it wasteful to run an inference model on an entire sequence at high output rates. Instead, change-driven sampling automatically identifies times when inference should be run.

Fig. 9 shows an example of change-driven sampling when reconstructing a balloon-burst sequence of 132000 frames. As the person moves the thumbtack to pop the balloon, the QNN outputs at 40 Hz. When the balloon ruptures, the QNN jumps as much as 24000 Hz, before dropping down to 200 Hz as the latex material flies out of the person’s hand. Overall, our quanta networks incur 127 inference steps, with a $150\times$ run-time speedup and $240\times$ less readout over inferring all outputs at 24 kHz. Change-driven output fidelity is also higher than uniform inference at comparable cost (running with $1024\times$ subsampling). As highlighted previously, we can combine change-driven sampling with eventful computation and readout; using a top-20% selection policy, the former reduces theoretical FLOP counts by $5\times$, while the latter reduces readout by $4.2\times$.

5.4. Is the Sensor-Proximal QNN Layer Sufficient?

In this section, we evaluate the effect of the location and nature of information aggregation performed by the QNN layers. We evaluate whether photon aggregation just at the sensor-proximal layer is sufficient or do later QNN layers increase performance and whether adaptive integration improves performance. In both ablations, for a fixed aggregator, we set $\omega_t = 255/256$ in Eq. (2). We measure performance across the intensity imaging, depth estimation, and point tracking tasks by training models in a similar manner as our overall approach. Further, for intensity imaging, we benchmark on the recently proposed i2k dataset [8] by interpolating ground truth frames from 2 kHz to 64 kHz and then simulating quanta frames at 0.3 PPP.

Sensor-proximal integrator type	Intermediate QNN Layers	Dense compute			Eventful compute (top-10%)		
		PSNR \uparrow dB	AJ \uparrow %	$\delta_{\text{acc}}\uparrow$ %	PSNR \uparrow dB	AJ \uparrow %	$\delta_{\text{acc}}\uparrow$ %
Fixed	\times	29.2	59.8	50.8	28.7	52.3	49.4
Adaptive	\times	30.3	60.1	53.2	29.9	54.6	52.1
Fixed	\checkmark	35.0	76.1	69.1	32.7	70.3	64.6
Adaptive	\checkmark	36.7	75.9	71.2	34.7	70.7	65.5

Table 1. **Ablation study.** We study the impact of including intermediate QNN layers and our choice of the sensor-proximal integrator. We see a marked improvement in performance when including learnable QNN layers (*top two vs bottom two rows*). Adaptive integration is less noisy than fixed integration that preserves comparable motion, resulting in better performance when using eventful computations (*right column*).

As Tab. 1 (*first row*) shows, using no adaptivity in temporal aggregation hampers task performance. Using just the sensor-proximal stage is a better choice, but is subpar to incorporating intermediate QNN layers. As the last two rows of Tab. 1 suggests, the general QNN layers can make up for performance in the absence of a first adaptive layer. However, when we apply eventful computations with a top-10% selection policy (as an example), fixed aggregation, which is noisier in slow-moving regions, leads to weaker performance than adaptive aggregation.

6. Limitations and Discussion

Quanta neural networks perform computer vision with photon detections and forego image reconstruction—with their efficient inference modes, this leads to over 2 orders of efficiency improvements over reconstruction-based quanta vision. However, our work is not without its limitations. The causal, point-wise QNN layers have less expressive temporal modeling than similar transformer layers. Eventful computations, while a promising pathway for reducing floating-point operations necessitate sparse workload backends for proportional run-time speedups.

Tasks limited by (labelled) dataset availability. We adopt a supervised training setup for QNNs in this work, which limits their application to computer-vision tasks where quanta-sensor datasets with paired groundtruth are available, possibly using simulation engines (*e.g.*, we use a Blender-based simulation pipeline for obtaining high-quality groundtruth depth and paired quanta-sensor data). To broaden the capabilities of quanta vision to tasks such as object detection and semantic segmentation, a self-supervised learning setup that can use unlabelled real-world quanta-sensor acquisition would be an enticing next step.

Blurring the sensing-processing boundary. As the affordances of near-sensor processors evolve, more QNN layers may be computed on sensor. Quanta neural networks of the future could operate in a split-computing setup [16, 17, 28, 42], with similar kinds of computations occurring before and after sensor readout—breaking down the distinction between upstream sensing and downstream processing. As an example of the ensuing implications, we could imagine sparse or eventful readout occurring at intermediate QNN layers; thus, the produced events would depart from encoding intensities to feature-map elements.

References

- [1] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007. [3](#), [12](#)
- [2] Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [16](#)
- [3] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240×180 130 dB 3 μ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. [5](#)
- [4] Claudio Bruschini, Harald Homulle, Ivan Michel Antolovic, Samuel Burri, and Edoardo Charbon. Single-photon spad imagers in biophotonics: Review and outlook. *arXiv preprint*, 2019. [1](#), [2](#)
- [5] Stanley H Chan. What does a one-bit quanta image sensor offer? *IEEE Transactions on Computational Imaging*, 8:770–783, 2022. [1](#)
- [6] Paramanand Chandramouli, Samuel Burri, Claudio Bruschini, Edoardo Charbon, and Andreas Kolb. A Bit Too Much? High Speed Imaging from Sparse Photon Counts. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, Tokyo, Japan, 2019. [1](#), [2](#)
- [7] Shiyang Chen, Chaoteng Duan, Zhaofei Yu, Ruiqin Xiong, and Tiejun Huang. Self-supervised mutual learning for dynamic scene reconstruction of spiking camera. In *IJCAI*, pages 2859–2866, 2022. [12](#)
- [8] Prateek Chennuri, Yiheng Chi, Enze Jiang, GM Godaliyadda, Abhiram Gnanasambandam, Hamid R Sheikh, Istvan Gyongy, and Stanley H Chan. Quanta video restoration. In *European Conference on Computer Vision*, pages 152–171. Springer, 2025. [1](#), [2](#), [6](#), [7](#), [8](#), [18](#), [19](#), [21](#)
- [9] Blender Online Community. Blender - a 3d modelling and rendering package, 2018. [16](#)
- [10] Tri Dao and Albert Gu. Transformers are ssm: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. [4](#), [12](#), [15](#)
- [11] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. [16](#)
- [12] Wenhao Dong, Haodong Zhu, Shaohui Lin, Xiaoyan Luo, Yunhang Shen, Xuhui Liu, Juan Zhang, Guodong Guo, and Baochang Zhang. Fusion-mamba for cross-modality object detection. *arXiv preprint arXiv:2404.09146*, 2024. [12](#)
- [13] Matthew Dutson, , Yin Li, and Mohit Gupta. Event neural networks. In *ECCV*, 2022. [5](#), [18](#)
- [14] Matthew Dutson, Yin Li, and Mohit Gupta. Eventful transformers: Leveraging temporal redundancy in vision transformers. In *ICCV*, pages 16911–16923, 2023. [13](#), [14](#)
- [15] Steinar Ekern. Adaptive exponential smoothing revisited. *Journal of the Operational Research Society*, 32(9):775–782, 1981. [3](#)
- [16] John Emmons, Sadjad Fouladi, Ganesh Ananthanarayanan, Shivararam Venkataraman, Silvio Savarese, and Keith Winstein. Cracking open the dnn black-box: Video analytics with dnns across the camera-cloud boundary. In *Proceedings of the 2019 workshop on hot topics in video analytics and intelligent edges*, pages 27–32, 2019. [8](#)
- [17] Amir Erfan Eshratifar, Amirhossein Esmaili, and Massoud Pedram. Bottlenet: A deep learning architecture for intelligent mobile cloud computing services. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6. IEEE, 2019. [8](#)
- [18] Eric R Fossum. What to do with sub-diffraction-limit (SDL) pixels?—a proposal for a gigapixel digital film sensor (DFS). In *IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, pages 214–217, 2005. [1](#)
- [19] Eric R Fossum, Jiaju Ma, Saleh Masoodian, Leo Anzagira, and Rachel Zizza. The quanta image sensor: Every photon counts. *Sensors*, 16(8):1260, 2016. [1](#)
- [20] Abhiram Gnanasambandam and Stanley H Chan. Image classification in the dark using quanta image sensors. In *ECCV*, pages 484–501. Springer, 2020. [2](#)
- [21] Abhiram Gnanasambandam, Omar Elgendy, Jiaju Ma, and Stanley H Chan. Megapixel photon-counting color imaging using quanta image sensor. *Optics express*, 27(12):17298–17310, 2019. [1](#)
- [22] Bhavya Goyal and Mohit Gupta. Photon-starved scene inference using single photon cameras. In *ICCV*, pages 2512–2521, 2021. [2](#), [19](#), [21](#)
- [23] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. [4](#), [12](#)
- [24] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. [12](#)
- [25] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *European conference on computer vision*, pages 222–241. Springer, 2024. [12](#)
- [26] Shantanu Gupta and Mohit Gupta. Eulerian Single-Photon Vision. In *ICCV*, 2023. [2](#)
- [27] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022. [7](#)
- [28] Diyi Hu and Bhaskar Krishnamachari. Fast and accurate streaming cnn inference via communication compression on the edge. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 157–163. IEEE, 2020. [8](#)
- [29] Ju Huang, Shiao Wang, Shuai Wang, Zhe Wu, Xiao Wang, and Bo Jiang. Mamba-fetrack: Frame-event tracking via state space model. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 3–18. Springer, 2024. [12](#)
- [30] Tiejun Huang, Yajing Zheng, Zhaofei Yu, Rui Chen, Yuan Li, Ruiqin Xiong, Lei Ma, Junwei Zhao, Siwei Dong, Lin Zhu, et al. 1000 \times faster camera and machine vision with ordinary devices. *Engineering*, 25:110–119, 2023. [12](#)
- [31] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pages 624–642. Springer, 2022. [16](#)
- [32] Atul Ingle, Andreas Velten, and Mohit Gupta. High Flux Passive Imaging With Single-Photon Sensors. In *CVPR*, 2019. [2](#)

- [33] Atul Ingle, Trevor Seets, Mauro Buttafava, Shantanu Gupta, Alberto Tosi, Mohit Gupta, and Andreas Velten. Passive inter-photon imaging. In *CVPR*, 2021. **2**
- [34] Kiyotaka Iwabuchi, Yusuke Kameda, and Takayuki Hamamoto. Image quality improvements based on motion-based deblurring for single-photon imaging. *IEEE Access*, 9:30080–30094, 2021. **2**
- [35] Sacha Jungerman, Atul Ingle, and Mohit Gupta. Panoramas from photons. In *ICCV*, pages 10626–10636, 2023. **2**
- [36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. **16**
- [37] Abhijit Kundu, Andrea Tagliasacchi, Anissa Yuenming Mak, Austin Stone, Carl Doersch, Cengiz Oztireli, Charles Herrmann, Dan Gnanapragasam, Daniel Duckworth, Daniel Rebain, et al. Kubric: A scalable dataset generator. 2022. **7, 16**
- [38] Martin Laurenzis, Trevor Seets, Emmanuel Bacher, Atul Ingle, and Andreas Velten. Comparison of super-resolution and noise reduction for passive single-photon imaging. *Journal of Electronic Imaging*, 31(3):033042–033042, 2022. **2**
- [39] Martin Laurenzis, Emmanuel Bacher, Trevor Seets, Atul Ingle, Andreas Velten, and Frank Christnacher. Single photon flux imaging with sub-pixel resolution by motion compensation. In *Advanced Photon Counting Techniques XVII*, pages 77–85. SPIE, 2023. **2**
- [40] Chengxi Li, Xiangyu Qu, Abhiram Gnanasambandam, Omar A. Elgendy, Jiaju Ma, and Stanley H. Chan. Photon-limited object detection using non-local feature matching and knowledge distillation. In *ICCVW*, pages 3976–3987, 2021. **2**
- [41] Dasong Li, Xiaoyu Shi, Yi Zhang, Ka Chun Cheung, Simon See, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. A simple baseline for video restoration with grouped spatial-temporal shift. In *CVPR*, pages 9822–9832, 2023. **6, 17**
- [42] Guangli Li, Lei Liu, Xueying Wang, Xiao Dong, Peng Zhao, and Xiaobing Feng. Auto-tuning neural network quantization framework for collaborative inference between the cloud and edge. In *International Conference on Artificial Neural Networks*, pages 402–411. Springer, 2018. **8**
- [43] Kunchang Li, Xinhao Li, Yi Wang, Yanan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pages 237–255. Springer, 2024. **12**
- [44] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. **6**
- [45] P Lichtsteiner. 64x64 event-driven logarithmic temporal derivative silicon retina. In *Program 2003 IEEE Workshop on CCD and AIS*, 2003. **5**
- [46] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2025. **12**
- [47] Jiaju Ma, Saleh Masoodian, Dakota A. Starkey, and Eric R. Fossum. Photon-number-resolving megapixel image sensor at room temperature without avalanche gain. *Optica*, 4(12):1474–1481, 2017. **1, 3**
- [48] Jiaju Ma, Dexue Zhang, Omar Elgendy, and Saleh Masoodian. A photon-counting 4mpixel stacked bsi quanta image sensor with 0.3 e-read noise and 100db single-exposure dynamic range. In *2021 Symposium on VLSI Circuits*, pages 1–2. IEEE, 2021.
- [49] Jiaju Ma, Dexue Zhang, Omar A Elgendy, and Saleh Masoodian. A 0.19 e-rms read noise 16.7 mpixel stacked quanta image sensor with 1.1 μm -pitch backside illuminated pixels. *IEEE electron device letters*, 42(6):891–894, 2021. **3**
- [50] Jiaju Ma, Stanley Chan, and Eric R Fossum. Review of quanta image sensors for ultralow-light imaging. *IEEE transactions on electron devices*, 69(6):2824–2839, 2022. **2**
- [51] Sizhuo Ma, Shantanu Gupta, Arin C. Ulku, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Quanta burst photography. *ACM TOG*, 39(4):1–16, 2020. **1, 2, 18, 19, 20**
- [52] Sizhuo Ma, Paul Mos, Edoardo Charbon, and Mohit Gupta. Burst vision using single-photon cameras. In *WACV*, pages 5375–5385, 2023. **1, 2**
- [53] Kazuhiro Morimoto, Andrei Ardelean, Ming-Lo Wu, Arin Can Ulku, Ivan Michel Antolovic, Claudio Bruschini, and Edoardo Charbon. Megapixel time-gated SPAD image sensor for 2D and 3D imaging applications. *Optica*, 7(4):346–354, 2020. **1**
- [54] K. Morimoto, J. Iwata, M. Shinohara, H. Sekine, A. Abdelghafar, H. Tsuchiya, Y. Kuroda, K. Tojima, W. Endo, Y. Maehashi, Y. Ota, T. Sasago, S. Maekawa, S. Hikosaka, T. Kanou, A. Kato, T. Tezuka, S. Yoshizaki, T. Ogawa, K. Uehira, A. Ehara, F. Inui, Y. Matsuno, K. Sakurai, and T. Ichikawa. 3.2 Megapixel 3D-stacked charge focusing spad for low-light imaging and depth sensing. In *2021 IEEE International Electron Devices Meeting (IEDM)*, pages 20.2.1–20.2.4, 2021. **1**
- [55] Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022. **12**
- [56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. **16**
- [57] Md Maklachur Rahman, Abdullah Aman Tutul, Ankur Nath, Lamyamba Laishram, Soon Ki Jung, and Tracy Hammond. Mamba in vision: A comprehensive survey of techniques and applications. *arXiv preprint arXiv:2410.03105*, 2024. **12**
- [58] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. **6, 16**
- [59] Hongwei Ren, Yue Zhou, Jiadong Zhu, Haotian Fu, Yulong Huang, Xiaopeng Lin, Yuetong Fang, Fei Ma, Hao Yu, and Bojun Cheng. Rethinking efficient and effective point-based networks for event camera classification and regression: Eventmamba. *arXiv preprint arXiv:2405.06116*, 2024. **12**
- [60] Alexis Rochas. Single photon avalanche diodes in cmos technology. Technical report, Citeseer, 2003. **1**
- [61] Trevor Seets, Atul Ingle, Martin Laurenzis, and Andreas Velten. Motion adaptive deblurring with single-photon cameras. In *WACV*, pages 1945–1954, 2021. **1, 2**
- [62] Yuan Shi, Bin Xia, Xiaoyu Jin, Xing Wang, Tianyu Zhao, Xin Xia, Xuefeng Xiao, and Wenming Yang. Vmambair: Visual state space model for image restoration. *IEEE Transactions on*

- Circuits and Systems for Video Technology*, 2025. [12](#)
- [63] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14489–14498, 2021. [7](#)
- [64] Jimmy T.H. Smith, Shalini De Mello, Jan Kautz, Scott Linderman, and Wonmin Byeon. Convolutional state space models for long-range spatiotemporal modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [12](#)
- [65] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. In *ICLR*, 2023. [12](#)
- [66] Varun Sundar, Andrei Ardelean, Tristan Swedish, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Sodacam: Software-defined cameras via single-photon imaging. In *ICCV*, pages 8165–8176, 2023. [2](#)
- [67] Varun Sundar, Matthew Dutton, Andrei Ardelean, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. Generalized event cameras. In *CVPR*, pages 25007–25017, 2024. [2](#), [5](#), [7](#), [12](#), [16](#), [18](#), [19](#)
- [68] DW Trigg and AG Leach. Exponential smoothing with an adaptive response rate. *Journal of the Operational Research Society*, 18(1):53–59, 1967. [3](#)
- [69] Arin Can Ulku, Claudio Bruschini, Ivan Michel Antolovic, Yung Kuo, Rinat Ankri, Shimon Weiss, Xavier Michalet, and Edoardo Charbon. A 512×512 SPAD Image Sensor With Integrated Gating for Widefield FLIM. *IEEE Journal of Selected Topics in Quantum Electronics*, 25(1):1–12, 2019. [1](#), [6](#)
- [70] Matthijs Van Keirsbilck, Alexander Keller, and Xiaodong Yang. Rethinking full connectivity in recurrent neural networks. *arXiv preprint arXiv:1905.12340*, 2019. [4](#)
- [71] Lishun Wang, Miao Cao, and Xin Yuan. Efficientsci: Densely connected network with space-time factorization for large-scale video snapshot compressive imaging. In *CVPR*, pages 18477–18486, 2023. [16](#), [18](#)
- [72] Zeyu Wang, Chen Li, Huiying Xu, and Xinzhong Zhu. Mamba yolo: Ssms-based yolo for object detection. *arXiv preprint arXiv:2406.05835*, 2024. [12](#)
- [73] Mian Wei, Sotiris Nousias, Rahul Gulve, David B. Lindell, and Kiriakos N. Kutulakos. Passive ultra-wideband single-photon imaging. In *ICCV*, pages 8135–8146, 2023. [1](#)
- [74] Rui Xu, Shu Yang, Yihui Wang, Yu Cai, Bo Du, and Hao Chen. Visual mamba: A survey and new outlooks. *arXiv preprint arXiv:2404.18861*, 2024. [12](#)
- [75] Feng Yang, Yue M. Lu, Luciano Sbaiz, and Martin Vetterli. Bits from photons: Oversampled image acquisition using binary poisson statistics. *IEEE Transactions on Image Processing*, 21(4):1421–1436, 2012. [3](#)
- [76] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. [2](#), [6](#)
- [77] Haobo Yuan, Xiangtai Li, Lu Qi, Tao Zhang, Ming-Hsuan Yang, Shuicheng Yan, and Chen Change Loy. Mamba or rwkv: Exploring high-quality and high-efficiency segment anything model. *arXiv preprint arXiv:2406.19369*, 2024. [12](#)
- [78] Jiyuan Zhang, Shanshan Jia, Zhaofei Yu, and Tiejun Huang. Learning temporal-ordered representation for spike streams based on discrete wavelet transforms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 137–147, 2023. [12](#)
- [79] Jinhang Zhang, Min Gao, Wenzhao Li, Dan Fang, and Chaowang Li. Visual state space model for image super-resolution. *IEEE Transactions on Instrumentation and Measurement*, 2024. [12](#)
- [80] Tianyi Zhang, Matthew Dutton, Vivek Boominathan, Mohit Gupta, and Ashok Veeraraghavan. Streaming quanta sensors for online, high-performance imaging and vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. [2](#)
- [81] Jing Zhao, Ruiqin Xiong, and Tiejun Huang. High-speed motion scene reconstruction for spike camera via motion aligned filtering. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020. [12](#)
- [82] Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2imgnet: Learning to reconstruct dynamic scene from continuous spike stream. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11996–12005, 2021. [12](#)
- [83] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. [2](#), [6](#), [16](#)
- [84] Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1432–1437. IEEE, 2019. [12](#)
- [85] Lin Zhu, Yunlong Zheng, Mengyue Geng, Lizhi Wang, and Hua Huang. Recurrent spike-based image restoration under general illumination. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8251–8260, 2023. [12](#)
- [86] Zeqi Zhu, Arash Pourtaherian, Luc Waeijen, Ibrahim Batuhan Akkaya, Egor Bondarev, and Orlando Moreira. Cats: Combined activation and temporal suppression for efficient network inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8166–8175, 2024. [5](#)
- [87] Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5819–5828, 2024. [12](#)