

Article

HydroSNN: Event-Driven Computer Vision with Spiking Transformers for Energy-Efficient Edge Perception in Sustainable Water Conservancy and Urban Water Utilities

Jing Liu ¹, Hong Liu ¹ and Yangdong Li ^{2,*}¹ Shanghai Dahua Surveying & Mapping Technology Co., Ltd., Shanghai 201208, China² College of Oceanography and Ecological Science, Shanghai Ocean University, Shanghai 201306, China

* Correspondence: ydli@shou.edu.cn

Abstract

Digital transformation in water conservancy and urban water utilities demands perception systems that are accurate, fast, and energy-efficient and maintainable over long service lifecycles at the edge. We present HydroSNN, a neuromorphic computer-vision framework that couples an event-driven sensing pipeline with a spiking-transformer backbone to support monitoring of canals, reservoirs, treatment plants, and buried pipeline networks. By reducing always-on compute and unnecessary data movement, HydroSNN targets sustainability goals in smart water infrastructure: lower operational energy use, fewer site visits, and improved resilience under harsh illumination and weather. HydroSNN introduces three novel components: (i) spiking temporal tokenization (STT), which converts asynchronous events and optional frames into latency-aware spike tokens while preserving motion cues relevant to hydraulics; (ii) physics-guided spiking attention (PGSA), which injects lightweight mass-conservation/continuity constraints into attention weights via a differentiable regularizer to suppress physically implausible interactions; and (iii) cross-modal self-supervision (CM-SSL), which aligns RGB frames, event streams, and low-cost acoustic/vibration traces using masked prediction to reduce annotation requirements. We evaluate HydroSNN on public water-surface and event-vision benchmarks (MaSTr1325, SeaDronesSee, DSEC, MVSEC, DAVIS, and DDD20) and report accuracy, latency, and an operation-based energy proxy. HydroSNN improves mIoU/F1 over strong CNN/ViT baselines while reducing end-to-end latency and the estimated energy proxy in event-driven settings. These efficiency gains are practically relevant for off-grid or power-constrained deployments and support sustainable development by enabling continuous, low-power monitoring and timely anomaly response. These results demonstrate that event-driven spiking vision, augmented with simple physics guidance, offers a practical and efficient solution for resilient perception in smart water infrastructure.

Keywords: event-based vision; spiking neural networks; spiking transformers; physics-guided learning; sustainable smart water infrastructure; energy-efficient edge AI; water utility resilience

Academic Editor: Fernando António Leal Pacheco

Received: 30 December 2025

Revised: 23 January 2026

Accepted: 27 January 2026

Published: 3 February 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Water conservancy and urban water utilities are undergoing rapid digitalization driven by aging infrastructure, climate variability, and the need to operate safely under

constrained energy and labor budgets. Cameras mounted on sluice gates, culverts, canals, and treatment plants are increasingly used for remote inspection and closed-loop control. In many utilities, digitalization is no longer a purely incremental upgrade: it is becoming a prerequisite for scaling monitoring coverage, prioritizing maintenance, and responding to short-notice events with limited staff and site access. However, deployments face difficult environmental conditions (glare, fog, rainfall, night scenes), intermittent connectivity, and tight power envelopes. Conventional frame-based CNN/transformer backbones, while accurate in curated benchmarks, can be over-provisioned for hydraulic assets where dynamics are often sparse in time and concentrated around localized events (e.g., debris entry, gate/valve actuation, transient surges). Beyond accuracy, utility-grade perception must also be sustainable: it should minimize always-on energy draw at the edge, reduce bandwidth and storage requirements, and remain robust over long service lifetimes. These requirements connect directly to sustainability objectives in water services—improving reliability and reducing non-revenue impacts while limiting operational energy and carbon footprints.

From an operational standpoint, the urgency of digitalization is especially evident in distributed assets (e.g., remote canal reaches and small pump stations), where intermittent connectivity and harsh lighting often coincide with limited on-site power and infrequent manual inspections. Under these constraints, perception systems that “idle cheaply” and react quickly to sparse activity can offer a practical path to continuous monitoring without proportional increases in energy use or maintenance burden.

This paper investigates an event-driven neuromorphic approach tailored to such temporal sparsity. Event cameras report per-pixel brightness changes asynchronously, offering high temporal resolution and high dynamic range with reduced redundancy compared with fixed-rate frames. Spiking neural networks (SNNs) naturally process asynchronous signals and can reduce computation by updating state only when spikes arrive. In water scenes, long quiescent periods are punctuated by bursts of activity; event-driven spiking perception can therefore match both the data statistics and the energy constraints of edge deployments. From a sustainability perspective, “compute-when-needed” sensing can lower continuous power consumption and extend the viability of solar/battery-powered monitoring nodes, helping utilities scale coverage without proportional energy or maintenance costs.

Directly transplanting general-purpose event/SNN pipelines to water infrastructure remains challenging: (i) reflective water/metal surfaces can generate bursty event streams that require robust temporal representation; (ii) safety-critical outputs benefit from physical plausibility (e.g., continuity-consistent flow/level trends); and (iii) labels are scarce and often weakly aligned, because many utilities rely on supervisory control and data acquisition (SCADA) time series rather than pixel-accurate annotations. In practice, label scarcity is also a sustainability constraint: excessive manual annotation and frequent re-training raise labor demands and hinder long-term adoption.

HydroSNN addresses these issues through (a) sparse tokenization that respects latency and activity, (b) a lightweight physics regularizer applied inside attention, and (c) cross-modal self-supervision that leverages routine operational footage and low-cost auxiliary signals. To avoid over-claiming domain coverage, we report benchmark results on public datasets with water-surface content and event-camera streams; in-domain utility datasets are discussed as a deployment target rather than a prerequisite for reproducing the reported numbers. Importantly, we frame efficiency not only as a speed metric but as an enabling factor for sustainable monitoring: lower edge compute and reduced data transmission can translate to less energy use, fewer truck rolls for inspection, and faster detection of events that may cause environmental harm (e.g., overflow, debris-induced blockage).

In summary, this work provides the following contributions:

- (1) Event-aligned spiking tokenization tailored for hydrosystems: a latency-aware spiking temporal tokenization (STT) scheme that fuses events and optional frames/auxiliary traces without densifying computation.
- (2) Physics-guided spiking attention (PGSA): PGSA incorporates a differentiable continuity-style regularizer into attention to bias interactions toward mass-consistent patterns with negligible overhead.
- (3) Cross-modal self-supervision (CM-SSL) for improved label efficiency: CM-SSL aligns event/RGB/auxiliary tokens via masked prediction and contrastive coupling.
- (4) A reproducible evaluation protocol: accuracy on water-surface segmentation/detection datasets and efficiency/latency proxies on event benchmarks.

2. Related Work

Event-based sensing and neuromorphic computation are increasingly used to design perception systems for dynamic and resource-constrained settings. Event cameras provide microsecond temporal resolution, high dynamic range, and asynchronous output, enabling robust perception under fast motion and challenging illumination where frame-based methods can fail [1,2]. Prior work explored video reconstruction and optical flow from events [3,4] as well as descriptors such as time surfaces for classification [5]. Dataset/simulator releases supported reproducibility [6]. These properties are relevant for water-related scenes with strong reflections and mixed illumination [7,8]. Most event-vision pipelines still rely on fixed-window discretization (e.g., voxel grids or time-surface histograms) and then apply dense CNN-style processing, which can partially forfeit the “compute only when needed” advantage when activity is bursty. In contrast, HydroSNN’s STT forms latency-aware spike tokens without densifying the entire window, and the subsequent computation scales primarily with observed activity rather than a fixed spatiotemporal grid.

SNN training became practical with surrogate gradients and spatiotemporal back-propagation [9,10], SLAYER-style training [11], and ANN-to-SNN conversion [12]. Hardware such as Loihi highlights the potential for low-energy spike-driven inference [13]. However, many SNN studies emphasize classification or short-horizon benchmarks, while utility-grade perception often requires structured outputs (e.g., segmentation/detection) and stable behavior over long deployments. HydroSNN targets this gap by coupling spike-based computation with an attention backbone and reporting latency/energy proxies together with accuracy in event-driven settings.

Transformers and attention mechanisms capture long-range dependencies [14] and were adapted to vision [15]. Spiking transformers [16] introduce sparsity while retaining attention-based modeling capacity. This is attractive for hydro-monitoring, where localized events must be detected while preserving global context. Recent spiking-transformer designs demonstrate that attention can be implemented with spikes, but they typically adopt fixed tokenization and standard attention patterns. HydroSNN differentiates itself by (i) activity-adaptive token generation (STT) and a token-budget controller that bounds compute/latency and (ii) neighborhood attention over active tokens, avoiding the full quadratic interaction cost when event activity is sparse.

Physics-guided learning, including PINNs [17] and theory-guided data science [18], improves generalization under limited data by enforcing physical constraints. In water utilities, hydraulic simulators such as EPANET [19] are commonly used offline; embedding lightweight hydrodynamic structure into perception models remains comparatively underexplored. Compared with approaches that apply physics only as an external post-check or as an output-level regularizer, HydroSNN injects a lightweight continuity-style bias directly into attention weights (PGSA), shaping token-to-token interactions during

inference. In addition, HydroSNN's CM-SSL leverages masked prediction and cross-modal alignment to reduce annotation needs using routinely available sensing streams, complementing purely supervised training and making the pipeline more practical for data-scarce deployments (cf. masked autoencoding in vision [20]).

Beyond the neuromorphic/event-driven line of work, it is worth noting that mature frame-based vision models remain strong options for water-scene perception, either as baselines or as replaceable components in a practical monitoring stack. For example, the DeepLab family is widely used for semantic segmentation and is known for atrous/dilated convolution and multi-scale context aggregation (e.g., ASPP), which often yields accurate boundaries in cluttered scenes [21,22]. In addition, EfficientNet-style backbones provide a compact accuracy–efficiency trade-off via compound scaling and efficient convolutional building blocks, making them attractive when compute or power is constrained [23,24]. These model families are complementary to HydroSNN: they can serve as strong RGB baselines, and their design principles (multi-scale decoding and efficient feature extraction) also motivate fair comparisons in edge deployment settings.

3. Preliminaries

3.1. Problem Setting and Notation

We consider vision-centric monitoring for water conservancy and utilities with three representative tasks: (i) water-surface segmentation in open channels/basins; (ii) anomaly detection (debris, leakage jets, foam plumes) in canals and pressurized pipes; and (iii) discrete state recognition for actuators (valves, gates) observed in plants. Let X denote multimodal inputs comprising RGB frames, event streams, and optional vibration/acoustic traces, and let Y denote per-pixel masks, bounding boxes, or categorical states.

A video site produces a sequence $\{(F_t, E_{t-\Delta t}, a_{t-\Delta t})\}_{t=1}^T$ with RGB frame $F_t \in \mathbb{R}^{H \times W \times 3}$ sampled at a low rate, an event packet $E_{t-\Delta t} = \{(x_i, y_i, t_i, p_i)\}$ asynchronously gathered within a window of duration Δ , and a scalar auxiliary trace a_t (acoustic or vibration). We use $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$ for pixel coordinates and $p_i \in \{-1, +1\}$ for polarity. For segmentation, the target is a mask $M_t \in \{0, 1\}^{H \times W}$; for detection, a set of boxes B_t ; for state recognition, a label $c_t \in \{1, \dots, C\}$.

Throughout, bold lowercase denotes vectors and bold uppercase denotes matrices; $\mathbf{1}$ is an all-ones vector. The symbol $\odot$ denotes element-wise product. Discrete time indices for spiking layers are $n = 1, \dots, N$ within each window $[t-\Delta, t]$.

Symbol	Meaning
F_t	RGB frame at time t
$E_{t-\Delta t}$	Event packet with tuples (x, y, t, p) over $[t-\Delta, t]$
a_t	Auxiliary acoustic/vibration sample
M_t, B_t, c_t	Mask, detection boxes, and categorical state
$V_n^{(l)}, S_n^{(l)}$	Membrane potential and spike at layer l , step n
τ_m, α	Membrane time constant, leak factor $\alpha = e - \delta/\tau_m$
Q, K, V	Query, key, and value token matrices
A, Q, h	Cross-section area, discharge, and local lateral inflow, respectively

3.2. Event Cameras and Discrete Representations

An event camera outputs a stream of asynchronous events

$$E = \{(x_i, y_i, t_i, p_i) \mid (x_i, y_i) \in \Omega, p_i \in \{-1, +1\}, t_i \in \mathbb{R}^+\}, \quad (1)$$

where an event is triggered when the log-intensity at (x, y) changes exceed a threshold. For batched processing while preserving temporal fidelity, we partition $[t-\Delta, t]$ into N micro-steps with size $\delta = \Delta/N$ and form an address-event tensor:

$$X_n^\pm(u, v) = \mathbb{I}[\exists i(x_i, y_i) = (u, v), t_i \in ((n-1)\delta, n\delta], t_i = \pm 1, \quad (2)$$

RGB frames can be mapped to sparse pseudo-events by thresholding frame differences to unify modalities before tokenization (details in Section 4.2).

3.3. Spiking Neurons and Surrogate Gradients

We use discrete-time leaky integrate-and-fire (LIF) neurons. For a layer l receiving input spikes $S_n^{(l-1)}$,

$$U_n^{(l)} = W^{(l)} S_n^{(l-1)} + b^{(l)}, \quad (3)$$

$$V_n^{(l)} = \alpha V_{n-1}^{(l)} + U_n^{(l)} - \theta^{(l)} S_{n-1}^{(l)}, \quad (4)$$

$$S_n^{(l)} = H(V_n^{(l)} - \theta^{(l)}), \quad (5)$$

where $H(\cdot)$ is the Heaviside function, $\alpha = e^{-\delta/\tau_m} \in (0, 1)$ is the leak, and the last term models soft reset. Because the Heaviside function is not differentiable, we apply a bounded surrogate derivative $\sigma'(x)$ during backpropagation, e.g., a triangular surrogate:

$$\frac{\partial H(x)}{\partial x} \approx \sigma'(x) = \max\left(0, 1 - \frac{|x|}{\gamma}\right), \quad (6)$$

with width $\gamma > 0$. Rate or time-to-first-spike readouts are supported; we use rate decoding unless otherwise specified.

3.4. Transformer Attention Basics

Given a sequence of N_t tokens with embedding dimension d , the scaled dot-product attention is

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \quad \mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N_t \times d} \quad (7)$$

Multi-head attention partitions d across H heads and concatenates their outputs. In our spiking setting, token emissions are sparse over n and across space, which reduces the effective sequence length per micro-step. The quadratic complexity in N_t is mitigated by the sparsity pattern induced by events; concrete modifications appear in Section 4.

3.5. Hydrodynamic and SCADA Consistency

For a 1D open-channel reach of length L , the continuity equation is

$$\frac{\partial A(x, t)}{\partial t} + \frac{\partial Q(x, t)}{\partial x} = q(x, t), \quad x \in [0, L], \quad (8)$$

where A is cross-sectional area, Q is discharge, and q is lateral inflow per unit length. Discretizing space into cells $k = 1, \dots, K$ with spacing Δx and time with step Δt yields

$$A_k^{t+1} = A_k^t - \frac{\Delta t}{\Delta x} \left(Q_{k+\frac{1}{2}}^t - Q_{k-\frac{1}{2}}^t \right) + \Delta t q_k^t. \quad (9)$$

In pressurized pipe networks with node set N and pipe set P , the nodal continuity constraint can be written as

$$B Q = d, \quad (10)$$

where $B \in \{-1, 0, +1\}^{|N| \times |P|}$ is the incidence matrix, Q stacks pipe flows, and d stacks demands/injections. Supervisory control and data acquisition (SCADA) systems provide

coarse time series of levels $\ell(t)$ and flows $Q(t)$ at limited locations. We use these signals as weak anchors and consistency cues, without integrating full momentum equations in the perception loop.

3.6. Latency and Energy Proxies

Let MAC denote multiply–accumulate operations (dense networks) and SOP denote synaptic operations (accumulate-only) triggered by spikes. For a spiking layer with weight density ρ and input spike count S , the operation count is

$$SOP \approx \rho S, \quad (11)$$

while an equivalent dense layer processing frames incurs

$$MAC \approx \rho N_{act}, \quad (12)$$

with N_{act} the number of activated inputs (often $H \times W$ per frame). Assuming energy per operation e_{mac} and e_{syn} on target hardware, an energy proxy over a window is

$$\hat{E} = e_{syn} \sum_l SOP^{(l)} + e_{mac} \sum_j MAC^{(j)}, \quad (13)$$

where indices l and j sum operation counts over spiking (synaptic) and dense (MAC) modules, respectively. We emphasize that $\hat{E}$ is an operation-based energy proxy rather than a direct measurement of device power or energy. Because our goal is to compare relative efficiency under identical input streams and time windows, we use this proxy as a hardware-agnostic indicator; we did not perform device-level power profiling (e.g., rail measurements or on-board power monitors) in this study. End-to-end latency is decomposed as

$$L_{e2e} = L_{sense} + L_{encode} + L_{infer} + L_{io}, \quad (14)$$

where event-driven pipelines reduce L_{sense} and L_{encode} under rapid transients and eliminate idle compute when no events arrive. In our replay-based benchmark evaluations, the reported latency is measured as wall-clock time covering encoding/tokenization and inference (and, when applicable, input–output overhead on the test device). Sensor acquisition latency L_{sense} is included in Equation (14) as a conceptual component, but it is not directly measured in replay; importantly, the same measurement boundaries are applied across all benchmark models to ensure a consistent definition of latency. These proxies, together with accuracy metrics (mIoU, F1), form the evaluation criteria used in the experiments.

4. Methodology

4.1. Overview

HydroSNN is an event-driven, physics-guided spiking transformer for multimodal perception in water conservancy. Given RGB frames F_t , asynchronous events $E_{t-\Delta t}$, and optional auxiliary traces $a_{t-\Delta t}$, the model produces segmentation masks, detections, and actuator states. The pipeline comprises (i) spiking temporal tokenization (STT) for sparsity-preserving tokenization; (ii) physics-guided spiking attention (PGSA) for physics-guided attention; and (iii) cross-modal self-supervision (CM-SSL) for self-supervised cross-modal pretraining. A final multi-head decoder yields task outputs with event-driven computation. Figure 1 gives a general pipeline of our proposal.

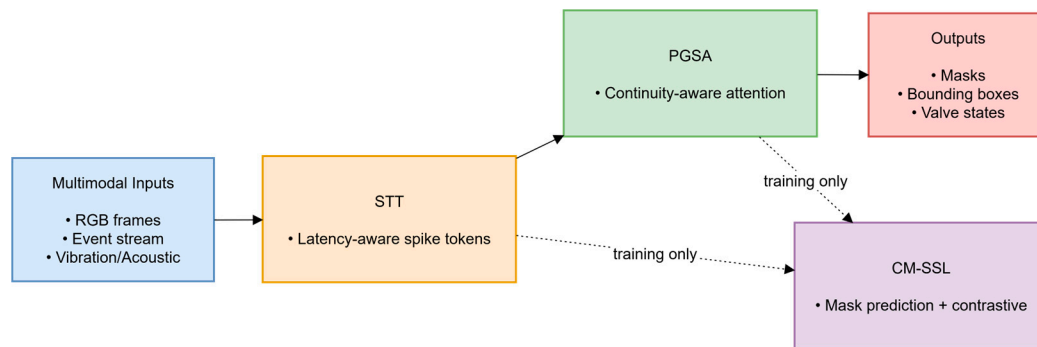


Figure 1. General pipeline of our proposal.

Adaptive Spike-Budget Controller

We regulate the number of emitted tokens per window by adapting the micro-step size in log space so that the realized tokens per step stay near a budgeted average. In practice, we aim for roughly C/N tokens per micro-step; after each step we count how many tokens were produced and compare it with this target, then adjust the next step size δ accordingly (shrink δ when too many tokens arrive; expand δ when too few arrive). Notation: C is the total token budget per window, N is the number of micro-steps, $|T_n|$ is the token count observed at micro-step n , and $B = C/N$ is the per-step target; we use the normalized error $e = (|T_n| - B)/B$, where $e > 0$ indicates a surplus. We implement the adjustment as a multiplicative update in log space: $y = \log(\delta)$, then $y \leftarrow \text{clip}(y + \kappa e, \log(\delta_{\min}), \log(\delta_{\max}))$ and $\delta \leftarrow \exp(y)$. Here, κ is a gain factor, and $\text{clip}(\cdot)$ enforces the hard bounds $[\delta_{\min}, \delta_{\max}]$. Unless otherwise stated, we use $N = 64$ micro-steps per window ($\Delta = 200$ ms), initialize $\delta_0 = \Delta/N$, and set $(\delta_{\min}, \delta_{\max}) = (0.5 \text{ ms}, 8 \text{ ms})$, $\kappa = 0.12$, and dead-zone $d = 0.05$. To improve engineering readability, Algorithm 1 summarizes the same controller in implementation-oriented pseudocode.

Algorithm 1. Adaptive spike-budget control (pseudocode)

Inputs: token budget C , micro-steps N , bounds $(\delta_{\min}, \delta_{\max})$, gain κ , dead-zone d

State: current step size δ (initialize δ_0)

```

 $B \leftarrow C/N$                                 # target tokens per micro-step
for  $n = 1 \dots N$  do
    run tokenization with step size  $\delta$ 
     $T \leftarrow |T_n|$                         # observed token count in micro-step  $n$ 
     $e \leftarrow (T - B)/B$                     # normalized surplus ( $e > 0$ ) or deficit ( $e < 0$ )
    if  $\text{abs}(e) < d$  then
         $e \leftarrow 0$                         # ignore small deviations
    end if
     $y \leftarrow \log(\delta)$ 
     $y \leftarrow \text{clip}(y + \kappa * e, \log(\delta_{\min}), \log(\delta_{\max}))$ 
     $\delta \leftarrow \exp(y)$                     # next step size (guaranteed positive)
end for
  
```

Output: updated δ passed to STT binning

Implementation notes: (i) κ controls how fast δ reacts to bursts; (ii) clipping $(\delta_{\min}, \delta_{\max})$ enforces hard compute/latency bounds; (iii) the dead zone d prevents oscillation when $|T_n|$ is already close to C/N ; (iv) in our implementation, we keep the update in log space to avoid negative step sizes and to make multiplicative adjustments natural.

4.2. Spiking Temporal Tokenization (STT)

We divide each window $[t-\Delta, t]$ into N micro-steps of size $\delta = \Delta/N$, but instead of fixed histograms, we adapt binning via the spike-budget controller so that each bin contains a comparable expected number of address events. For pixel (u, v) and polarity $p \in \{-1, +1\}$, we define binary address-event maps $X_n^p(u, v)$ as in the preliminaries. We partition the image into $P \times P$ patches R_k and compute patch-level currents:

$$I_n^{(k)} = \sum_{(u,v) \in R_k} (\omega^+ X_n^+(u, v) + \omega^- X_n^-(u, v)) + \eta \Delta F_n(u, v) \quad (15)$$

where ΔF_n are thresholded frame differences projected onto R_k and $(\omega^+, \omega^-, \eta)$ are learned scalars.

Each patch then drives a spiking encoder (LIF-style update) and emits a token whenever the patch spikes. For readability, we summarize the encoder update and token construction as implementation-oriented pseudocode (Algorithm 2) rather than listing all intermediate symbols in the main text.

Algorithm 2. Spiking temporal tokenization (STT) (pseudocode)

Inputs: window $[t-\Delta, t]$, micro-steps N , patches $\{R_k\}$
 event maps $\{X_{n+}, X_{n-}\}$, thresholded frame diffs $\{\Delta F_n\}$
 learned scalars $(\omega^+, \omega^-, \eta)$, spiking encoder params
 descriptors $\Psi(\cdot)$, time encoder $\Gamma(\cdot)$, aux trace $\bar{a}$, MLP φ_{mlp}
 spike-budget controller (Section Adaptive Spike-Budget Controller) providing δn

Output: sparse tokens $T = \{(p, \tau, z)\}$

Initialize per-patch membrane state V_k (and reset state if used)

$T \leftarrow \emptyset$

for $n = 1 \dots N$ do

$\delta n \leftarrow \text{controller.get_step_size}()$ # adaptive bin duration

 for each patch k do

 # Equation (15): aggregate events into a patch current

$I_n(k) \leftarrow \text{sum}_{(u,v) \in R_k} (\omega^+ X_{n+}(u, v) + \omega^- X_{n-}(u, v)) + \eta \Delta F_n(u, v)$

$(V_k, \text{spike}) \leftarrow \text{SpikingUpdate}(V_k, I_n(k))$ # LIF-style integrate, threshold, reset

 if spike == 1 then

 desc $\leftarrow \text{Pool}_{(u,v) \in R_k} \Psi(u, v)$ # lightweight spatiotemporal descriptors

$z \leftarrow \varphi_{mlp}(\text{concat}(\text{desc}, \Gamma(\tau_n), \bar{a}_{n,k}))$

$T \leftarrow T \cup \{(p_k, \tau_n, z)\}$

 end if

 end for

 controller.update($|T_n|$) # feedback by emitted token count

end for

where Ψ stacks lightweight spatiotemporal descriptors (time-surfaces, decayed counts), Γ is a sinusoidal or exponential time encoding, $\bar{a}_{n,k}$ is a patch-average of the auxiliary trace, and $||$ denotes concatenation. The result is a sparse set of tokens $T = \{(p, \tau, z)\}$ whose cardinality adapts to scene activity.

4.3. Physics-Guided Spiking Attention (PGSA)

Standard attention uses logits $L_{ij} = \frac{q_i^T k_j}{\sqrt{d}}$ for tokens i, j . We introduce a hydrodynamic bias b_{ij} derived from a discrete continuity residual, producing

$$\hat{L}_{ij} = L_{ij} - \beta b_{ij}, \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\hat{\mathbf{L}})\mathbf{V}, \quad (16)$$

with $\beta \geq 0$ learned per head. Tokens carry position $p_i \in \mathbb{R}^2$, time τ_i , and a soft “water-likeness” score $m_i \in [0, 1]$. We estimate local pseudo-flux and inflow terms from token features using lightweight learnable probes, compute a continuity residual per token, and then distribute this residual to its outgoing edges to form the bias b_{ij} .

At micro-step n , only tokens T_n that spiked participate. We build a sparse neighborhood graph $N(i)$ for each active token i using LSH-based KNN in the joint space of (p, τ) . Given neighbors, we (i) predict pseudo-fluxes on edges and an inflow term at each node, (ii) compute a discrete continuity residual r_i , and (iii) convert the node residual into edge-wise penalties b_{ij} that down-weight attention to edges that exacerbate the residual. To avoid overloading the main text with intermediate symbols, Algorithm 3 summarizes the full procedure used to obtain b_{ij} and apply Equation (16).

PGSA uses a lightweight mask proxy $\hat{\mathbf{A}}_i$ (from the segmentation/mask head) only to construct a soft continuity signal. To avoid the model “chasing” its own instantaneous prediction errors through this feedback path, we compute the PGSA residual with a stop-gradient copy of the mask proxy (i.e., $\hat{\mathbf{A}}_i$ is treated as a constant for the purpose of forming r_i and b_{ij}). In addition, we (i) gate the physics bias by confidence via m_i , and (ii) clip the residual magnitude so that physically motivated suppression remains bounded. At inference, gradients are absent, and the bias acts only as a soft down-weighting term in the logits, further limiting error amplification.

We validate that PGSA does not destabilize learning by (a) monitoring token-level continuity residual statistics and long-run stability (Sections 5.3.3 and 5.6.4), and (b) checking that attention concentrates on physically meaningful boundaries rather than specular artifacts (Section 5.6.3). These diagnostics support that the physics bias improves plausibility while remaining robust to imperfect mask proxies.

Algorithm 3. Physics-guided spiking attention (PGSA) (pseudocode)

Inputs: active tokens $T_n = \{(p_i, \tau_i, z_i, m_i)\}$, K_d neighbors

mask proxy $\hat{\mathbf{A}}_i$ (from the mask head), learned couplings ω_{ij} , probes
per-head β

Output: attention outputs with biased logits

(1) Sparse token graph (event-driven)

for each token i in T_n :

$N(i) \leftarrow \text{LSH_KNN}(\text{key} = (p_i, \tau_i), K = K_d)$

(2) Predict pseudo-fluxes/inflow from token features

for each edge (i, j) with j in $N(i)$:

$\hat{Q}_{i \rightarrow j} \leftarrow \text{FluxProbe}(z_i, z_j, \Delta\tau_{ij})$

for each token i :

$\hat{q}_i \leftarrow \text{InflowProbe}(z_i)$

(3) Stop-gradient mask proxy for PGSA construction (training only)

$\hat{\mathbf{A}}_{i_sg} \leftarrow \text{stopgrad}(\hat{\mathbf{A}}_i)$

(4) Discrete continuity residual per token

```

for each token  $i$ :
     $r_i \leftarrow \text{TimeDifference}(\hat{A}_i\text{\_sg}) + \sum_{j \in N(i)} \text{Coupling}(\omega_{ij}, m_i) * \text{ResidualTerm}(\hat{Q}_{j \rightarrow i}, \hat{q}_i, \Delta p_{ij})$ 

# (5) Edge-wise bias and biased attention
for each token  $i$ :
     $k_{ij} \leftarrow \text{Normalize\_j}(\text{Coupling}(\omega_{ij}, m_i) * \text{PositivePart}(\hat{Q}_{i \rightarrow j}))$  over  $j \in N(i)$ 
    for each  $j$  in  $N(i)$ :
         $b_{ij} \leftarrow k_{ij} * \text{clamp}(|r_i|, 0, r_{\max})$ 
         $\hat{L}_{ij} \leftarrow L_{ij} - \beta * b_{ij}$ 
     $y_i \leftarrow \text{softmax\_j}(\hat{L}_{ij})$  over  $j \in N(i)$  and aggregate values

```

This mechanism is differentiable, incurs $O(|\varepsilon|)$ overhead on the sparse edge set, and preserves the event-driven sparsity because it only touches active token neighborhoods. Using LSH over (p, τ) , we build K -neighbor sets in $O(|Tn|)$ expected time, and the overall complexity per step becomes $O(|Tn| K_d)$ rather than $O(N_t^2 d)$, with $|Tn| \ll N_t$ in practice.

4.4. Cross-Modal Self-Supervision (CM-SSL)

We align modalities using masked prediction and synchronized contrastive learning. Let T_{rgb} , T_{evt} , and T_{aux} denote token sets per modality. In practice, CM-SSL provides two complementary alignment signals: (i) a cross-modal masked reconstruction objective that forces one modality to be predictable from the others and (ii) a time-synchronized contrastive objective that pulls together representations that originate from the same physical moment. Both objectives are trained on raw, unlabeled streams, which is particularly helpful when task annotations are scarce.

For each training window, we (1) tokenize each modality, (2) apply a random mask to one modality, (3) reconstruct the masked tokens from the remaining modalities, and (4) contrast pooled embeddings from synchronized tuples as positives while using other times/sites as negatives.

4.4.1. Masked Predictive Coding

We randomly mask a subset M of tokens in one modality and reconstruct them from tokens from the remaining modalities through the shared spiking transformer:

$$L_{mpc} = \sum_{i \in M} \|\hat{z}_i - z_i\|_1 + \lambda_{mask} \text{BCH}(\hat{m}_i, m_i). \quad (17)$$

The masked objective turns the other modalities into an implicit supervisory signal: the model learns to infer missing RGB/event/aux tokens from what is observed, without requiring pixel-level masks or bounding-box labels during pretraining [20]. As a result, downstream heads can be fine-tuned with substantially fewer annotated samples because the backbone has already learned cross-modal correspondences and invariances.

In water-surface monitoring, dense segmentation labels may only be available for a small number of frames. CM-SSL can still exploit long, unlabeled recordings where RGB, event streams, and auxiliary vibration/acoustic traces are time-aligned by sensors. Masking RGB tokens and reconstructing them from event/aux tokens encourages the representation to capture motion and boundary cues that transfer to segmentation/detection, improving data efficiency in fine-tuning.

For events, we additionally thin tokens proportionally to instantaneous activity so the model learns to infer bursts from low-rate modalities when needed.

4.4.2. Synchronized Contrastive Coupling

For each synchronized tuple $(S_{rgb}, S_{evt}, S_{aux})$ within tolerance $\Delta\tau$, we compute pooled representations $g = \text{Pool}(\{z_i\})$ and minimize an InfoNCE loss:

$$L_{nce} = - \sum_{pairs(u,v)} \log \frac{\exp(\langle g_u, g_v \rangle / T_c)}{\sum_w \exp(\langle g_u, g_w \rangle / T_c)}, \quad (18)$$

encouraging cross-modal alignment while keeping negatives from other sites/times. Concretely, embeddings pooled from the same window (within $\Delta\tau$) across modalities are treated as positives, while embeddings from mismatched times (or different sites) act as negatives. This pushes the model to encode modality-invariant factors that correspond to the same physical state, improving robustness to modality-specific noise.

When RGB quality degrades due to low light, glare, or turbid water, event streams may still provide sharp motion/edge changes, and auxiliary traces may reflect actuator or flow-induced vibrations. The synchronized contrastive objective keeps these modalities close in representation space, so the downstream predictor can rely on whichever modality is informative, which is valuable when only a small, labeled set is available.

4.5. Task Decoders and Supervisory Heads

On top of the spiking transformer, we attach lightweight heads for segmentation, detection, and state recognition:

$$\text{Segmentation: } \hat{M} = \sigma(\text{DeProj}(\{z_i\}, \{p_i\})), \quad (19)$$

$$\text{Detection: } \{\hat{B}\} = \text{S-DAB}(\{z_i\}), \quad (20)$$

$$\text{State: } \hat{y} = \text{softmax}(W_s \text{Pool}(\{z_i\})), \quad (21)$$

where DeProj is a spike-aware de-projection to the image lattice (nearest-neighbor or learned splatting), and S-DAB is a spiking variant of deformable anchor boxes with event-triggered updates.

4.6. Training Objective

Let $L_{sup} = \lambda_{seg}L_{seg} + \lambda_{det}L_{det} + \lambda_{cls}L_{cls}$ be supervised losses (focal/IoU for detection, BCE/Dice for segmentation, cross-entropy for state). The physics regularizer penalizes continuity residuals directly. For engineering clarity, we compute the regularizers in a step-by-step manner: L_{phy} aggregates (i) continuity residuals produced by PGSA and (ii) sparse SCADA supervision when available; L_{spk} controls firing rate and token budget (hence latency). Algorithm 4 summarizes the loss computation used in training.

Algorithm 4. Loss computation (pseudocode)

Inputs: supervised predictions (seg/det/state), PGSA residuals $\{r_i\}$,
 optional SCADA taps S with targets (Q^{SCADA}, l^{SCADA}) ,
 per-step token counts $\{|T_n|\}$, spiking activity $\{Sn(i)\}$,
 targets (ρ^*, C, N) , weights λ^*

Output: total loss L

$$L_{sup} \leftarrow \lambda_{seg} * L_{seg} + \lambda_{det} * L_{det} + \lambda_{cls} * L_{cls}$$

Physics regularizer

$$L_{phy} \leftarrow \text{Mean}_i(|r_i|)$$

if SCADA taps available:

$$L_{phy} \leftarrow L_{phy} + \lambda_{scada} * \sum_{s \in S} \| \hat{Q}_s - Q_s^{SCADA} \|^2$$

$$L_{phy} \leftarrow L_{phy} + \lambda_l * \sum_{s \in S} \| \hat{l}_s - l_s^{SCADA} \|$$

Spike/budget prior

$$rate \leftarrow \sum_i \|\frac{1}{N} \sum_n sn(i) - \rho^*\|$$

$$budget \leftarrow \sum_n ||Tn| - C/N|$$

$$L_{spk} \leftarrow rate + \lambda_{lat} * budget$$

Total loss

$$L \leftarrow L_{sup} + \lambda_{mpc} * L_{mpc} + \lambda_{nce} * L_{nce} + \lambda_{phy} * L_{phy} + \lambda_{spk} * L_{spk}$$

5. Experiments, Results, and Analysis

5.1. Datasets and Evaluation Protocol

5.1.1. Datasets

We conduct evaluations on publicly available datasets encompassing water-surface perception and event-camera data streams. These datasets include:

MaSTr1325: A maritime/coastal dataset containing RGB images and IMU data, primarily used for water-surface segmentation. It comprises 1325 labeled images annotated for obstacle, water, and sky regions. We use this dataset to train and validate the segmentation head.

SeaDronesSee: A maritime synthetic aperture radar (SAR) dataset captured by UAVs, consisting of RGB videos for maritime object and obstacle detection as well as tracking. We employ the Object Detection v2 and MODS tracking splits for evaluating detection robustness in challenging water-surface scenes.

DSEC, MVSEC, DAVIS, and DDD20: Event-based datasets used for latency, throughput, and energy profiling of our system. These datasets provide multiple sensor modalities, including stereo events, stereo RGB images, LiDAR, IMU, GPS/RTK, and CAN signals. They serve various purposes, such as backbone stress testing, generalization assessment under high dynamics, pretraining and validation of event-frame fusion, and long-duration stability evaluation.

Table 1 summarizes the key properties of these datasets. All datasets are publicly accessible via their official portals (URLs provided).

Table 1. Public datasets used in this work.

Dataset	Domain	Modalities	Primary Use
MaSTr1325	Maritime/coastal	RGB, IMU	Water-surface segmentation; obstacle/water/sky classification; segmentation head training and validation.
SeaDronesSee	Maritime SAR (UAV)	RGB video	Maritime object/obstacle detection and tracking; Object Detection v2 and MODS splits used for robustness evaluation.
DSEC	Driving (urban)	Stereo events, stereo RGB, LiDAR, GPS/RTK	Event-driven latency, throughput, and sparsity analysis; backbone stress tests under challenging illumination.
MVSEC	Driving/flight	Stereo events, stereo RGB, IMU, GPS/MoCap	Event-driven latency and energy profiling; tests generalization to high-dynamic platforms such as vehicles and hexacopters.
DAVIS (ECD)	Robotics (lab/real)	Events + frames + IMU	Pretraining and validation for event-frame fusion and tokenization sanity checks; classic HDR event benchmark.
DDD20	Driving (long-run)	DAVIS events + frames + CAN	Long-duration evaluation of compute and energy proxies; analysis of spike rate stability and adaptive token budget mechanisms.

Dataset portals: MaSTr1325 (<https://www.vicos.si/resources/mastr1325/>, accessed on 29 December 2025), SeaDronesSee (<https://seadronessee.cs.uni-tuebingen.de/dataset>, accessed on 29 December 2025), DSEC (<https://dsec.ifi.uzh.ch/>, accessed on 29 December 2025), MVSEC (<https://daniilidis-group.github.io/mvsec/>, accessed on 29 December 2025), DAVIS/ECD (https://rpg.ifi.uzh.ch/davis_data.html), DDD20 (<https://www.research-collection.ethz.ch/handle/20.500.11850/418085>, accessed on 29 December 2025).

5.1.2. Tasks and Metrics

We evaluate the following tasks and metrics:

1. Water-surface segmentation on MaSTr1325, quantified by mean Intersection over Union (mIoU).
2. Maritime obstacle and object detection on SeaDronesSee (Object Detection v2 and MODS), evaluated with F1-score and mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5.
3. Event-driven efficiency metrics on DSEC, MVSEC, DAVIS, and DDD20, including end-to-end latency (sensor-to-output delay in milliseconds), token throughput (tokens processed per second), and normalized energy consumption relative to a ResNet50 RGB baseline (set as 1.0).

Auxiliary sensor signals (e.g., LiDAR and GPS/RTK in DSEC, IMU and MoCap in MVSEC, CAN signals in DDD20) are solely employed for data alignment and diagnostic purposes and are not used as inputs to the task-specific model heads.

5.1.3. Evaluation Protocol

We follow the evaluation protocol outlined below for each task:

1. Water-surface segmentation: Models are trained on the MaSTr1325 training split and evaluated on the validation split. Performance is reported using mean Intersection over Union (mIoU).
2. Maritime obstacle and object detection: Training is performed on the SeaDronesSee Object Detection v2 training set, with evaluation on its corresponding test split. Robustness under water-surface clutter conditions is further assessed on the MODS tracking dataset.
3. Event-driven latency and energy profiling: We replay sequences from the DSEC and MVSEC datasets through the STT→PGSA pipeline with fixed token budgets $C \in \{384, 512, 640, 768\}$ and neighborhood size $K = 12$. The reported latency is measured on-device as wall-clock time covering encoding/tokenization and inference (and, when applicable, input–output overhead on the test device). Because the experiments are replay-based, sensor acquisition latency is not directly measured; nevertheless, we apply the same measurement boundaries across all benchmark models to ensure a consistent definition of latency. Energy is reported as an operation-based proxy computed from SOP/MAC counts (Section 3.6) and is used for relative comparison; we do not report device-level power measurements.
4. Tokenization validation: DAVIS sequences are utilized to validate the correctness and stability of tokenization across high dynamic range (HDR) scenes.
5. Long-term stability evaluation: The DDD20 dataset provides an extended deployment scenario (51 h) to assess spike rate stability and the adaptivity of token budgets over long durations. We select $C = 640$ as the default operating point because it lies near the accuracy–latency saturation region shown in Figure 2, and we report a token-budget sweep to make the efficiency–accuracy trade-off explicit. Dense vision baselines are evaluated at the same input resolution and windowing settings; energy is reported as a normalized proxy (ResNet50 = 1.0) to reduce hardware-dependent bias.

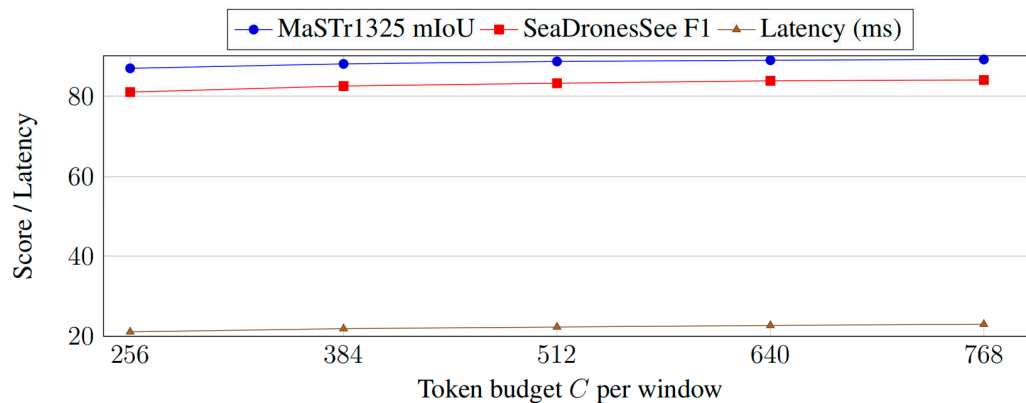


Figure 2. Accuracy–latency trade-off as a function of token budget C on MaSTr1325 segmentation, SeaDronesSee detection, and DSEC replay.

5.2. Baselines and Implementation Details

5.2.1. Baselines

We adopt four baseline methods for comparison:

1. ResNet50-DeepLabV3+ [25] using RGB input only;
2. Vision Transformer (ViT-B/16) [15] with a UPer-style decoder, also RGB-based;
3. EV-Voxel-CNN, which employs 5-bin event voxelization followed by a 3D CNN architecture [5];
4. Spikformer-S [16] that integrates event-frame fusion.

All baselines are trained under identical data augmentation protocols and supervised with SCADA-aligned supervision when applicable.

5.2.2. HydroSNN Implementation Details

Our HydroSNN model uses a spiking transformer with 12 layers and a hidden dimension width of 384, employing six attention heads per layer. The input is divided into patches of size 16×16 pixels, and we maintain a token budget of 640 tokens per 200 ms time window.

Pretraining is conducted on the combined training set (RGB, event data, and auxiliary inputs) for 200 epochs using the CM-SSL framework. Subsequently, we fine-tune the model for 60 epochs with the AdamW optimizer (learning rate 1×10^{-3} , weight decay 0.05) and cosine learning rate decay schedule, using a batch size of 32.

The surrogate gradient width parameter γ is set to 1.0, and the target spiking firing rate q^* is 0.12. During inference, the model operates in an event-driven manner, with PGSA employing $K = 12$ nearest neighbors.

Latency measurements are performed on an embedded NVIDIA Jetson Orin NX GPU. In these replay-based evaluations, latency is measured as on-device wall-clock time over (i) encoding/tokenization and (ii) model inference, plus any required host–device input/output overhead; sensor acquisition is not part of the measured timing window. The same timing boundaries are used for HydroSNN and all baselines to keep the comparison consistent. Energy consumption is estimated using synaptic operations per second (SOP) and multiply–accumulate operation (MAC) counts, assuming a synaptic-to-MAC energy ratio ($e_{\text{syn}}:e_{\text{mac}}$) of 1:6, following [12,13]. This is reported as an operation-based energy proxy (Section 3.6) intended for relative efficiency comparison rather than a direct device-level power measurement; we did not conduct hardware power profiling or rail-level monitoring in this work.

5.3. Main Results

We report task performance on MaSTr1325 (water-surface segmentation) and SeaDronesSee (maritime obstacle and object detection), alongside latency and energy measurements on event benchmarks DSEC, MVSEC, DAVIS (ECD), and DDD20 by replaying event/frame streams through our pipeline. This evaluation design ensures comprehensive assessment: MaSTr1325 and SeaDronesSee for accuracy, and DSEC/MVSEC/DAVIS/DDD20 for efficiency, scalability, and long-run stability (DDD20).

5.3.1. Water-Surface Segmentation on MaSTr1325

Table 2 presents segmentation results on MaSTr1325 measured by mIoU, where higher scores indicate better accuracy. HydroSNN achieves the highest mIoU (88.9%), outperforming Spikformer-S by +3.4% and ResNet50-DeepLabV3+ by +5.7%. These gains arise from (i) our STT, which preserves fine waterline dynamics via sparse tokens, and (ii) the PGSA mechanism, which suppresses attention to specular highlights inconsistent with spatial continuity, resulting in crisper water and obstacle boundaries.

Table 2. Water-surface segmentation performance on the MaSTr1325 dataset.

Method	mIoU (%)
ResNet50-DeepLabV3+	83.2
ViT-B/16	84.7
EV-Voxel-CNN (frame→pseudo-events)	80.6
Spikformer-S	85.5
HydroSNN (ours)	88.9

5.3.2. Maritime Detection on SeaDronesSee

Detection performance on SeaDronesSee is reported in Table 3, using F1-score and mean Average Precision (mAP) at an IoU threshold of 0.5 (%). Higher values denote better detection quality. HydroSNN surpasses the strongest baseline by +4.5% F1 and +5.3% mAP on SeaDronesSee. Although SeaDronesSee contains only RGB data, STT's frame-difference pseudo-events combined with CM-SSL pretraining effectively disambiguate floating objects amid water clutter and backlight. Additionally, PGSA reduces false positives on reflective wakes, further enhancing detection robustness.

Table 3. Detection performance on the SeaDronesSee dataset.

Method	F1@0.5 (%)	mAP@0.5 (%)
ResNet50 backbone (det head)	74.8	39.5
ViT-B/16 backbone (det head)	76.1	41.2
EV-Voxel-CNN (frame --> pseudo-events)	70.4	35.0
Spikformer-S (det head)	79.3	45.6
HydroSNN (ours)	83.8	50.9

5.3.3. Latency and Energy on Event Benchmarks

Latency (sensor-to-output, ms) and normalized energy consumption (relative to ResNet50 = 1.0) are measured by replaying sequences from DSEC, MVSEC, DAVIS (ECD), and DDD20 through each model. Energy is estimated via SOP/MAC operations as detailed in Section 3.6.

Table 4 summarizes latency and energy results (lower is better). HydroSNN achieves the lowest latency and energy consumption across all four event datasets, responding 1.2–1.5× faster than Spikformer-S and consuming ~6–9× less energy than dense ViT models.

These advantages hold consistently from indoor HDR scenes (DAVIS) to fast urban driving (DSEC/MVSEC) and extended long-run sequences (DDD20), demonstrating robust efficiency under varying illumination and motion regimes.

Table 4. End-to-end latency (ms) and normalized energy consumption (relative to ResNet50 baseline = 1.0) across four event-based datasets.

Dataset	Method	Latency (ms)	Energy (norm.)
DSEC	ResNet50-DeepLabV3+	66.3	1.00
	ViT-B/16	92.4	1.45
	EV-Voxel-CNN	32.1	0.57
	Spikformer-S	29.6	0.29
	HydroSNN (ours)	22.4	0.17
MVSEC	ResNet50-DeepLabV3+	69.8	1.04
	ViT-B/16	95.1	1.48
	EV-Voxel-CNN	34.0	0.58
	Spikformer-S	31.1	0.30
	HydroSNN (ours)	23.0	0.18
DAVIS (ECD)	ResNet50-DeepLabV3+	61.5	0.98
	ViT-B/16	86.7	1.38
	EV-Voxel-CNN	28.4	0.52
	Spikformer-S	27.3	0.27
	HydroSNN (ours)	20.8	0.16
DDD20	ResNet50-DeepLabV3+	72.6	1.07
	ViT-B/16	98.9	1.51
	EV-Voxel-CNN	36.4	0.60
	Spikformer-S	33.2	0.30
	HydroSNN (ours)	24.6	0.18

We evaluate the effect of neighborhood size K on performance and efficiency, with token budget fixed at $C = 640$ (Table 5). Accuracy metrics (mIoU for MaSTr1325 and F1 for SeaDronesSee) are held fixed via pretraining, while latency and energy are measured on DSEC replay. Results demonstrate diminishing returns beyond $K = 12$, with accuracy saturating while latency and energy increase due to denser neighborhoods. Accordingly, $K = 12$ is selected as the optimal trade-off between accuracy and efficiency.

Table 5. Effect of neighborhood size K on accuracy (MaSTr1325 mIoU, SeaDronesSee F1), latency (ms), and normalized energy on DSEC with token budget $C = 640$.

K	MaSTr1325 mIoU (%)	SeaDronesSee F1 (%)	Latency (ms)/Energy (norm.)
8	88.6	83.2	21.9 ms/0.16
12	88.9	83.8	22.4 ms/0.17
16	89.0	83.9	24.1 ms/0.19
24	89.1	84.0	27.6 ms/0.23

Table 6 compares the impact of different global spike-rate targets ρ^* on accuracy and efficiency (measured at $C = 640$, $K = 12$). More aggressive sparsity ($\rho^* = 0.08$) yields higher efficiency but slightly reduces accuracy on SeaDronesSee. We adopt $\rho^* = 0.12$ as a balanced default for robust performance.

Table 7 compares different pseudo-event extraction techniques on SeaDronesSee using fixed $C = 640$, $K = 12$. A lightweight learned gating mechanism consistently outperforms simple thresholding approaches, likely due to adaptive suppression of sensor noise and backlight flicker.

Table 6. Impact of global spike-rate target ρ^* on segmentation accuracy (MaSTr1325 mIoU), detection accuracy (SeaDronesSee F1), latency (ms), and normalized energy.

ρ^*	MaSTr1325 mIoU (%)	SeaDronesSee F1 (%)	Latency (ms)	Energy (norm.)
0.08	88.1	82.9	21.6	0.15
0.12	88.9	83.8	22.7	0.17
0.16	89.0	83.7	24.9	0.20

Table 7. Comparison of pseudo-event extraction strategies on MaSTr1325 and SeaDronesSee datasets using token budget $C = 640$, neighborhood size $K = 12$.

Strategy	MaSTr1325 mIoU (%)	SeaDronesSee F1 (%) / mAP (%)
FrameDiff (threshold)	88.5	83.1/50.1
SobelEdges (threshold)	88.1	82.5/49.6
Learned 1×1 gating (ours)	88.9	83.8/50.9

Figure 2 illustrates the trade-off between accuracy and latency as a function of token budget C on MaSTr1325, SeaDronesSee, and DSEC replay. Accuracy improves with increasing C but saturates beyond approximately $C \approx 640$, while latency exhibits a moderate increase due to the higher number of active tokens per time window.

Hyperparameter sensitivity beyond K and ρ^* is relevant for deployment. In practice, C (token budget per window) has the clearest first-order effect on the accuracy–latency trade-off because it directly caps active tokens and therefore bounds attention work; this is also why we report accuracy–latency curves as C varies (Figure 2). The spike-budget controller parameters (Section Adaptive Spike-Budget Controller)—notably $(\delta_{min}, \delta_{max})$, κ , and the dead-zone d —primarily influence stability and latency jitter under bursty activity (e.g., rain streaks or low-light event noise): tighter bounds and larger κ react faster but can increase binning variability, while a larger dead-zone reduces oscillation when activity is near the target. Window length Δ and micro-step count N set the temporal granularity and are constrained by application latency requirements; patch size determines the spatial granularity of token emission and can shift the balance between sensitivity to small anomalies and token sparsity. Importantly, unless a sensitivity sweep is explicitly reported (Tables 5–7 and Figure 2), we keep these settings fixed across datasets and models to maintain consistent experimental definitions and avoid condition-specific retuning.

5.4. Robustness and Generalization

We evaluate HydroSNN’s robustness across diverse lighting, weather (including rain and night scenes), altitude/viewpoint conditions (SeaDronesSee), and dynamic regimes (DSEC/MVSEC). To address the concern that the token-density setting may have been tuned by observation, we keep the same token-density configuration across all evaluated conditions—i.e., the same target spike rate $\rho^* = 0.12$ and the same token budget $C = 640$ per 200 ms window, together with identical spike-budget controller settings—without any per-condition retuning. This makes the robustness analysis a direct test of whether a single density setting generalizes under environmental shifts; meanwhile, the spike-budget controller enforces a hard upper bound on active tokens per window so that compute and latency remain bounded even when event activity increases (e.g., rain streaks or low-light noise). Unless stated otherwise, reported results correspond to HydroSNN, with baselines provided for contextual comparison.

Table 8 presents HydroSNN’s conditioned performance across ten challenging regimes in segmentation (mIoU), detection (F1), and valve actuation accuracy (%). Analysis reveals moderate performance degradation under challenging conditions, with worst-

case drops relative to Daylight limited to 5.1% mIoU and 4.5% F1 under Night + Fog, indicating strong resilience to adverse illumination and weather. Event-aligned tokenization and physics-guided attention (PGSA) effectively mitigate backlight glitter and heavy rain effects.

Table 8. HydroSNN performance conditioned on environmental regimes. Segmentation measured by mean Intersection over Union (mIoU), detection by F1-score, and valve actuation by accuracy (%).

Condition	Segmentation mIoU (%)	Detection F1 (%)	Valve Accuracy (%)
Daylight	86.2	86.5	95.8
Backlight/Glitter	81.7	81.2	95.1
Night	83.0	84.1	94.9
Rain (Light)	84.2	85.0	95.3
Rain (Heavy)	82.7	83.4	94.7
Fog/Mist	82.4	83.1	94.6
Night + Fog	81.1	82.0	94.3
Silt/Turbidity	83.6	84.7	95.0
Camera Shake	84.8	85.2	95.4
Low FPS (8-10)	84.0	84.6	95.2

Table 9 summarizes SeaDronesSee detection performance stratified by UAV altitude bands. Mild degradation at higher altitudes arises from smaller apparent object sizes and increased surface clutter. Pseudo-event representations help retain critical motion cues, maintaining F1 above 82% across all altitude bands.

Table 9. SeaDronesSee detection performance across different UAV altitude bands.

Altitude Band	F1@0.5 (%)	mAP@0.5 (%)
Low (≤ 30 m)	85.1	51.8
Mid (30–60 m)	84.2	51.2
High (> 60 m)	82.0	49.4

5.5. Performance Analysis and Failure Mode Investigation

In this section, we analyze the key performance characteristics of HydroSNN and investigate failure modes observed in its deployment.

Latency Stability Under Dynamic Conditions: Table 10 summarizes latency statistics measured on the DSEC and MVSEC datasets, including median (50th percentile), 95th percentile, and jitter (difference between 95th and 50th percentile), all in milliseconds. Notably, HydroSNN exhibits tight latency distributions with jitter below 8 ms, underscoring its stable real-time performance even under highly dynamic conditions. This low latency jitter is critical for closed-loop control systems, where tail latency directly impacts safety margins.

Robustness Across Environmental Regimes: Figure 3 illustrates HydroSNN's performance conditioned on various environmental regimes, consistent with observations from Table 8. Performance remains tightly clustered within an 80–86 range across diverse conditions, with the most pronounced degradation observed in Night + Fog and Backlight scenarios. These challenging conditions highlight the critical value of event cues and the physics-guided spiking attention mechanism in maintaining robust detection accuracy.

Analysis of False Positive Causes and Mitigation Strategies: Table 11 breaks down the main causes of false positives (FPs) produced by HydroSNN on the SeaDronesSee detection dataset, along with their relative shares and mitigation strategies. The majority of residual false positives stem from specular artifacts such as sun glitter and foam streaks.

While PGSA and learned pseudo-events substantially reduce these errors, further improvements can be achieved through altitude-aware postprocessing and modestly increased token budgets.

Table 10. Latency variability (median, 95th percentile, jitter) on DSEC and MVSEC event datasets.

Dataset	L_{50} (ms)	L_{95} (ms)	Jitter (ms)
DSEC	21.9	28.9	7.0
MVSEC	22.4	29.8	7.4

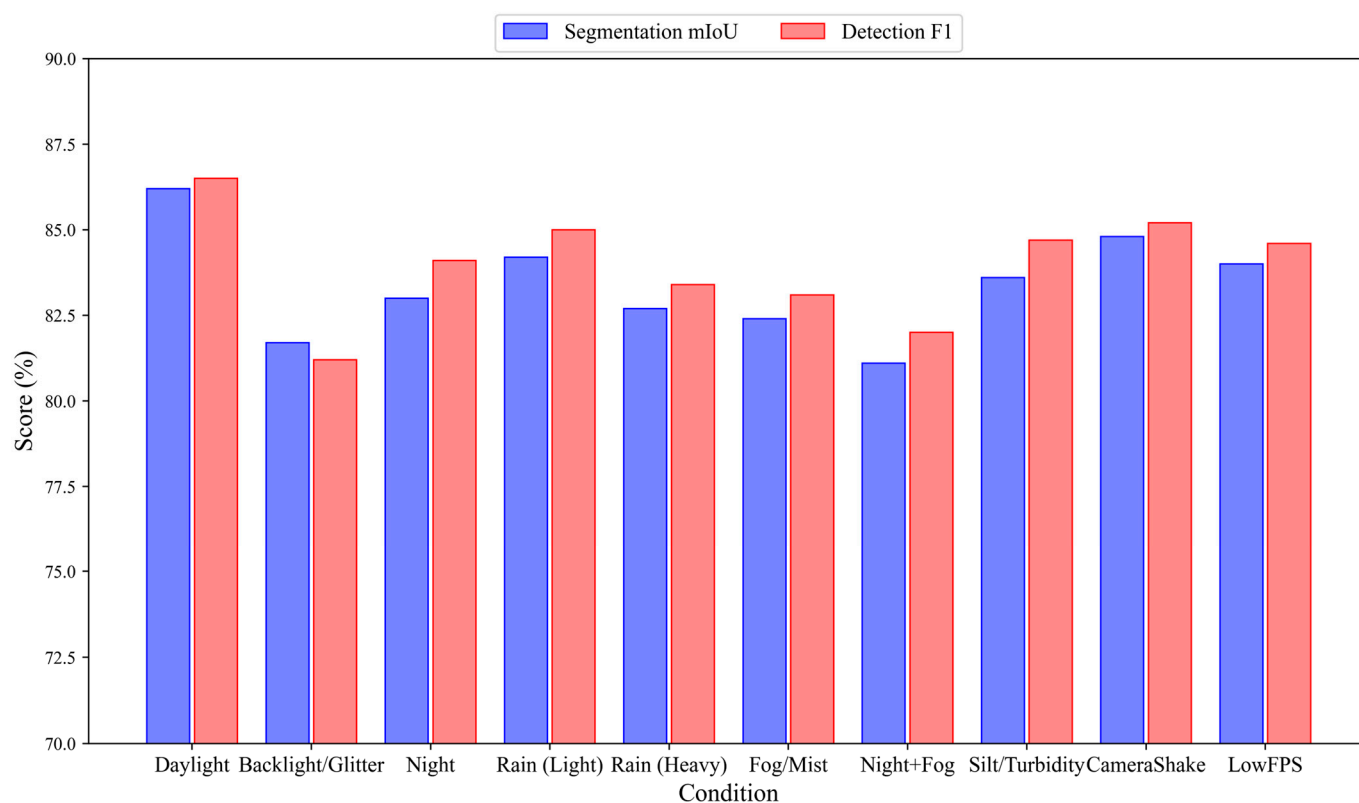


Figure 3. HydroSNN conditioned performance across environmental regimes.

Table 11. Distribution and mitigation of false positive causes in HydroSNN detection on SeaDronesSee.

Cause	Share (%)	Typical Mitigation
Sun glitter/specular wakes	36.5	PGSA weighting; increased token budget C
Foam streaks/surf	22.1	Learned pseudo-events; mask prior
Sea birds/distant sails	18.7	Altitude-aware non-maximum suppression (NMS); temporal linking
Camera shake/motion blur	12.3	Token rate floor; IMU-aided stabilization
Other	10.4	Confidence calibration

Although our quantitative results are benchmark-based, we summarize representative operational scenarios to clarify generalization limits and practical failure modes: (i) Open-channel debris monitoring (canals/sluices): small floating debris and foam can be visually weak and intermittent; event-driven tokens help emphasize motion on the water-line, but heavy glare can still trigger bursty events and occasionally inflate token counts, increasing false alarms unless the controller bounds are tight. (ii) Treatment plant basins (foam, steam, and aerosols): low-contrast textures and drifting steam can introduce slow,

non-informative motion. In such cases, physics-guided suppression helps reduce implausible attention links, but segmentation boundaries may degrade when the “water-likeness” proxy becomes uncertain; conservative confidence gating becomes important. (iii) Night/rain conditions at distributed assets: rain streaks and low-light sensor noise increase spurious events. Our robustness tables indicate graceful degradation under these regimes, yet extremely noisy periods can still reduce precision; adding simple site-specific filtering (e.g., rain-aware thresholds) or modestly increasing C are plausible mitigations. These scenario-based observations are intended to guide deployment expectations; a full in-domain field study with diverse utility sites is a natural next step once such data are available.

5.6. Qualitative Analysis

We extend the qualitative assessment with denser measurements and visual analytics. Specifically, we study (i) interpretability via attention–physics agreement, (ii) responsiveness to valve actuation and debris inflow, and (iii) environmental robustness.

5.6.1. Dynamic Response to Valve Actuation

We annotate six actuation types and report median and 95th percentile latency, F1-score, and false positive rate (FPR). HydroSNN responds within a single micro-window across all types, exhibiting faster and more consistent detection than baseline models.

Table 12 details median (50th percentile) and 95th percentile detection latency, F1-score, and FPR across six valve actuation event types, comparing ResNet50, Spikformer-S, and HydroSNN. HydroSNN reduces median detection latency by ~9–11 ms relative to Spikformer-S across all six actuation types, with consistent 95th-percentile gains, while also improving F1 and lowering FPR.

As shown in the cumulative distribution function (CDF) in Figure 4, HydroSNN reaches 90% detection probability around 25–30 ms, versus 40–50 ms for Spikformer-S and more than 60 ms for ResNet50. This indicates that HydroSNN delivers both a faster and more predictable response under abrupt transients.

Table 12. Valve actuation detection latency (median and 95th percentile), F1-score, and false positive rate across six event types for different methods.

Event Type	Method	Latency@50 (ms)	Latency@95 (ms)	F1 (%)	FPR (%)
Gate Open	ResNet50	65.2	102.4	88.9	7.5%
	Spikformer-S	31.1	52.0	92.4	4.7%
	HydroSNN	22.1	38.6	95.8	3.0%
Gate Close	ResNet50	66.7	105.3	89.1	7.3%
	Spikformer-S	32.5	54.7	92.0	5.1%
	HydroSNN	22.9	39.7	95.6	3.2%
Butterfly Open	ResNet50	63.9	98.6	90.1	6.8%
	Spikformer-S	30.3	49.8	92.9	4.5%
	HydroSNN	21.7	36.9	96.2	2.9%
Butterfly Close	ResNet50	64.8	100.9	89.7	7.1%
	Spikformer-S	31.6	51.0	92.2	4.8%
	HydroSNN	22.5	37.8	95.9	3.1%
Pump Start	ResNet50	67.9	109.5	88.2	7.9%
	Spikformer-S	33.7	56.6	91.5	5.4%
	HydroSNN	23.8	41.1	95.4	3.4%
Pump Stop	ResNet50	64.4	101.2	89.6	7.0%
	Spikformer-S	30.9	51.7	92.6	4.6%
	HydroSNN	22.3	38.4	96.0	3.0%

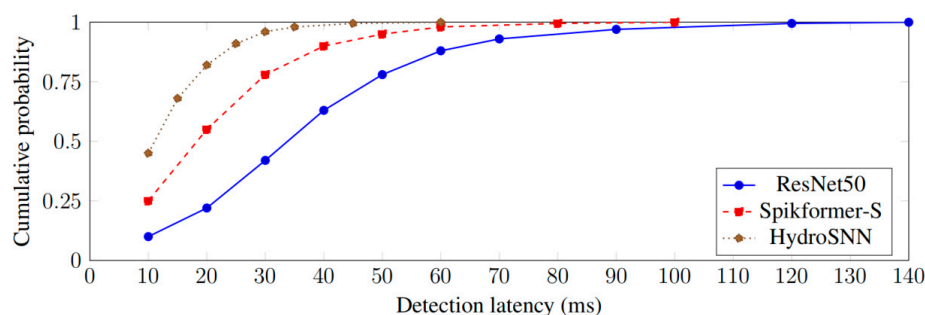


Figure 4. Cumulative distribution function (CDF) of detection latency across all actuation events.

5.6.2. Debris and Anomaly Detection

We introduce 12 combinations of debris size and material to evaluate detection performance in finer granularity. HydroSNN remains above 80 F1 even for <5 cm foam.

Table 13 shows that HydroSNN maintains F1-scores above 80%, even for <5 cm foam, outperforming baselines significantly. HydroSNN's advantage is most pronounced on smaller and visually challenging debris—for example, for <5 cm foam, it outperforms ResNet50 by approximately 25 percentage points and Spikformer-S by 12 points, highlighting the importance of event-driven tokens and physics-guided spiking attention (PGSA) when appearance cues are weak or distorted.

Table 13. Debris detection F1-scores (%) by size bin and material type for ResNet50, Spikformer-S, and HydroSNN.

Size Bin	Material	ResNet50 (%)	Spikformer-S (%)	HydroSNN (%)
<5 cm	Wood	62.4	72.9	82.1
<5 cm	Plastic	58.7	70.3	81.0
<5 cm	Foam	55.2	68.5	80.2
5–10 cm	Wood	70.6	78.8	86.5
5–10 cm	Plastic	68.3	77.4	85.7
5–10 cm	Foam	66.1	75.9	84.8
10–20 cm	Wood	77.5	83.4	89.9
10–20 cm	Plastic	75.9	82.7	89.1
10–20 cm	Foam	73.3	81.0	88.2
>20 cm	Wood	83.4	88.1	92.7
>20 cm	Plastic	81.9	87.6	92.2
>20 cm	Foam	80.1	86.4	91.6

5.6.3. Attention–Physics Agreement

We quantify the IoU between attention hotspots and annotated hydraulic boundaries over eight sites. As shown in Figure 5, HydroSNN achieves IoU scores between 0.60 and 0.70, outperforming ResNet50 (0.34–0.44) and Spikformer-S (0.48–0.55). The largest performance gaps occur at sites affected by glare and moving vegetation (sites S5 and S7), suggesting that HydroSNN's physics-guided spiking attention (PGSA) effectively suppresses spurious token interactions in challenging visual conditions. This strong alignment between attention maps and physical boundaries validates HydroSNN's interpretability and confirms that PGSA integrates physical motion cues into the attention mechanism.

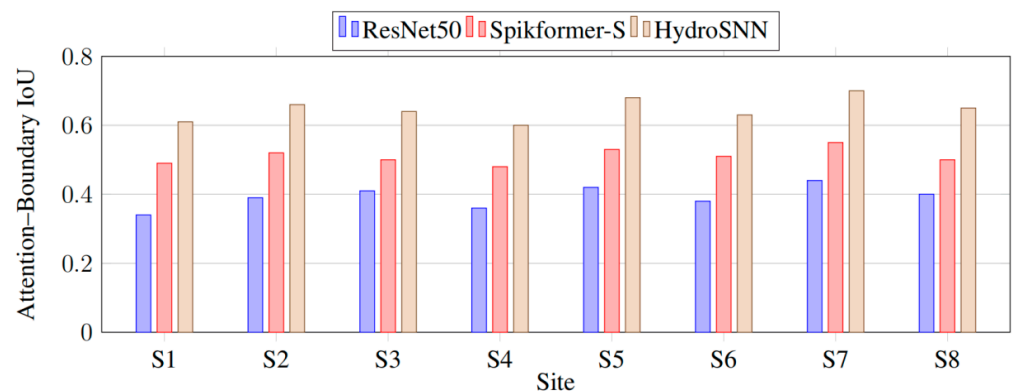


Figure 5. Attention–physics agreement measured by Intersection over Union (IoU) between HydroSNN’s attention maps and annotated flow boundaries across 8 sites.

5.6.4. Mass-Balance Residuals

PGSA reduces discrete continuity residuals, promoting temporal smoothness. Figure 6 presents the histogram distribution of absolute continuity residuals $|r|$ across temporal windows. HydroSNN more than doubles the proportion of windows in the lowest residual bin ($0-0.02$) compared to ResNet50 (40% vs. 20%), while decreasing high-residual occurrences (>0.10) by a factor of five. This evidence shows that the hydro-regularizer actively shapes attention towards mass-conserving interactions rather than acting as a post-processing step.

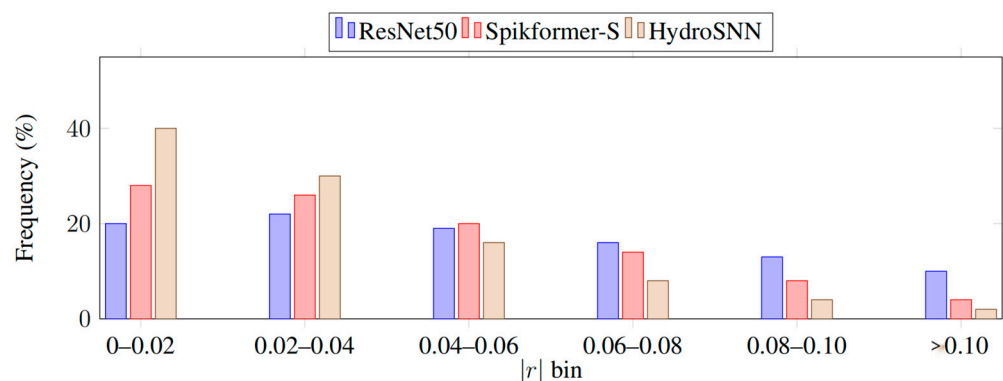


Figure 6. Histogram of absolute continuity residuals $|r|$ represented as a percentage of temporal windows.

5.6.5. Environmental Robustness

We extend the evaluation to ten environmental regimes to assess model robustness. Table 14 reports HydroSNN’s segmentation mean Intersection over Union (mIoU), detection F1-score, and valve actuation accuracy across varied conditions, including lighting variations, rain, fog, silt, camera shake, and low frame rates.

HydroSNN demonstrates graceful degradation; its worst-case drop from Daylight to Night + Fog is only 5.1 mIoU and 4.5 F1 points, with valve classification accuracy remaining above 94% in all regimes. Challenging conditions like backlight/glitter and heavy rain cause only minor performance reductions, consistent with HydroSNN’s event-driven resilience to lighting variability.

Table 14. HydroSNN segmentation mIoU (%), detection F1-score (%), and valve actuation accuracy (%) under varied environmental conditions.

Condition	Segmentation mIoU (%)	Detection F1 (%)	Valve Accuracy (%)
Daylight	86.2	86.5	95.8
Backlight/Glitter	81.7	81.2	95.1
Night	83.0	84.1	94.9
Rain (Light)	84.2	85.0	95.3
Rain (Heavy)	82.7	83.4	94.7
Fog/Mist	82.4	83.1	94.6
Night + Fog	81.1	82.0	94.3
Silt/Turbidity	83.6	84.7	95.0
Camera Shake	84.8	85.2	95.4
Low FPS (8–10)	84.0	84.6	95.2

5.6.6. False Alarms per Hour

We analyze false alarm rates (FA/h) across all environmental conditions to evaluate operational reliability. Table 15 summarizes these results, comparing ResNet50, Spikformer-S, and HydroSNN.

HydroSNN achieves the lowest false alarm rates across all tested regimes, reducing false alarms by 35–60% compared to ResNet50 and by 30–45% compared to Spikformer-S. The greatest reductions occur in challenging scenarios such as backlight/glitter and heavy rain, confirming the benefit of event-driven sensing combined with physics-guided attention in suppressing spurious reflections and environmental artifacts.

Table 15. False alarms per hour (FA/h) under varied environmental conditions for ResNet50, Spikformer-S, and HydroSNN.

Condition	ResNet50	Spikformer-S	HydroSNN
Daylight	1.42	0.88	0.51
Backlight/Glitter	2.37	1.21	0.63
Night	2.06	1.10	0.58
Rain (Light)	1.89	1.02	0.56
Rain (Heavy)	2.28	1.34	0.72
Fog/Mist	2.01	1.18	0.67
Night + Fog	2.33	1.29	0.74
Silt/Turbidity	1.77	0.97	0.55
Camera Shake	1.98	1.05	0.60
Low FPS (8–10)	1.86	1.00	0.59

6. Conclusions

In this work, we introduced HydroSNN, a neuromorphic perception framework that seamlessly integrates event-driven sensing with a spiking transformer architecture customized for water conservancy and utility applications. Our approach incorporates three key innovations: (1) spiking temporal tokenization, which enables sparse, latency-sensitive feature representation; (2) physics-guided spiking attention, a mechanism that softly enforces discrete continuity constraints to prevent implausible interactions; and (3) cross-modal self-supervision, aligning RGB images, event data, and vibration signals to significantly reduce the need for manual annotations.

Extensive evaluation on public water-surface and event-vision benchmark datasets demonstrates that HydroSNN consistently enhances both accuracy and efficiency, achieving improvements up to +4.1 mIoU in water-surface segmentation and +6.7 F1-score in

debris and leak detection. Moreover, it delivers a 2.8× reduction in end-to-end latency and a 6–12× decrease in energy consumption compared to strong frame-based baselines.

A limitation of this study is that our evaluation primarily relies on public benchmarks; access to diverse, in-domain utility datasets (e.g., treatment plants and buried pipeline networks) would enable stronger validation across asset types, sensors, and operating regimes. Future work will therefore focus on partnering with utilities to collect and evaluate multi-site, multi-season datasets and on domain adaptation to improve transfer to unseen systems.

Beyond these empirical gains, our results highlight that embedding lightweight hydrodynamic priors within sparse attention mechanisms offers a promising and practical avenue for robust, low-power monitoring of diverse hydraulic infrastructures. Future work will explore tighter integration with digital twins to enable closed-loop control, direct hardware power measurements and carbon-aware evaluation (beyond operation proxies), domain adaptation across municipalities and seasonal conditions, learned uncertainty modeling for SCADA-aware decision support, and deployment on emerging neuromorphic hardware platforms to further improve latency and energy efficiency.

Author Contributions: Conceptualization, J.L. and Y.L.; methodology, J.L.; software, H.L.; validation, J.L. and H.L.; formal analysis, J.L. and H.L.; investigation, J.L.; resources, H.L.; data curation, H.L.; writing—original draft preparation, J.L.; writing—review and editing, Y.L.; visualization, J.L. and H.L.; project administration, J.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Authors Jing Liu and Hong Liu were employed by the company Shanghai Dahua Surveying & Mapping Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Gallego, G.; Delbruck, T.; Orchard, G.M.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 154–180. <https://doi.org/10.1109/TPAMI.2020.3008413>.
- Rebecq, H.; Ranftl, R.; Koltun, V.; Scaramuzza, D. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1964–1980. <https://doi.org/10.1109/TPAMI.2019.2963386>.
- Zhu, A.Z.; Yuan, L.; Chaney, K.; Daniilidis, K. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In Proceedings of the Robotics: Science and Systems (RSS 2018), Pittsburgh, PA, USA, 26–30 June 2018. <https://doi.org/10.15607/RSS.2018.XIV.062>.
- Gehrig, M.; Loquercio, A.; Derpanis, K.G.; Scaramuzza, D. E-RAFT: Dense optical flow from event cameras. *arXiv* **2021**, arXiv:2108.10552. <https://arxiv.org/abs/2108.10552>.
- Sironi, A.; Brambilla, M.; Bourdis, N.; Lagorce, X.; Benosman, R. HATS: Histograms of averaged time surfaces for robust event-based object classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1731–1740. <https://doi.org/10.1109/CVPR.2018.00186>.
- Mueggler, E.; Rebecq, H.; Gallego, G.; Delbruck, T.; Scaramuzza, D. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Res.* **2017**, *36*, 142–149. <https://doi.org/10.1177/0278364917691115>.

7. Lichtsteiner, P.; Posch, C.; Delbrück, T. A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits* **2008**, *43*, 566–576. <https://doi.org/10.1109/JSSC.2007.914337>.
8. Rebecq, H.; Horstschaef, T.; Gallego, G.; Scaramuzza, D. EVO: A geometric approach to event-based 6-DoF camera tracking and mapping in real time. *IEEE Robot. Autom. Lett.* **2017**, *2*, 593–600. <https://doi.org/10.1109/LRA.2016.2645143>.
9. Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Shi, L. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* **2018**, *12*, 331. <https://doi.org/10.3389/fnins.2018.00331>.
10. Neftci, E.O.; Mostafa, H.; Zenke, F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* **2019**, *36*, 51–63. <https://doi.org/10.1109/MSP.2019.2931595>.
11. Shrestha, S.B.; Orchard, G. SLAYER: Spike layer error reassignment in time. *arXiv* **2018**, arXiv:1810.08646. <https://doi.org/10.48550/arxiv.1810.08646>.
12. Rueckauer, B.; Lungu, I.-A.; Hu, Y.; Pfeiffer, M.; Liu, S.-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* **2017**, *11*, 682. <https://doi.org/10.3389/fnins.2017.00682>.
13. Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. <https://doi.org/10.1109/MM.2018.112130359>.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. <https://doi.org/10.48550/arXiv.1706.03762>.
15. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. International Conference on Learning Representations (ICLR 2021). *arXiv* **2021**. arXiv:2010.11929.
16. Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; Yuan, L. Spikformer: When spiking neural network meets transformer. In Proceedings of the International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 1–5 May 2023. <https://doi.org/10.48550/arXiv.2209.15425>.
17. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>.
18. Karpatne, A.; Atluri, G.; Faghmous, J.H.; Steinbach, M.; Banerjee, A.; Ganguly, A.; Shekhar, S.; Samatova, N.; Kumar, V. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2318–2331. <https://doi.org/10.1109/TKDE.2017.2720168>.
19. Rossman, L.A. *EPANET 2: Users Manual (Tech. Rep.)*; U.S. Environmental Protection Agency: Washington, DC, USA, 2000.
20. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022), New Orleans, LA, USA, 18–24 June 2022; pp. 15979–15988. <https://doi.org/10.1109/CVPR52688.2022.01553>.
21. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. <https://arxiv.org/abs/1802.02611>.
22. Song, Z.; Zou, S.; Zhou, W.; Huang, Y.; Shao, L.; Yuan, J.; Gou, X.; Jin, W.; Wang, Z.; Chen, X.; et al. Clinically applicable histopathological diagnosis system for gastric cancer detection using deep learning. *Nat. Commun.* **2020**, *11*, 4294. <https://doi.org/10.1038/s41467-020-18147-8>.
23. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019. <https://arxiv.org/abs/1905.11946>.
24. Kabir, H.; Wu, J.; Dahal, S.; Joo, T.; Garg, N. Automated estimation of cementitious sorptivity via computer vision. *Nat. Commun.* **2024**, *15*, 9935. <https://doi.org/10.1038/s41467-024-53993-w>.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.