Spiking transformer with learnable threshold mechanism for underwater image dehazing to aid vision-based navigation

NEUROMORPHIC
Computing and Engineering

OPEN ACCESS

RECEIVED
5 December 2025

REVISED
25 January 2026

ACCEPTED FOR PUBLICATION
4 February 2026

PUBLISHED
17 February 2026

PAPER

Spiking transformer with learnable threshold mechanism for underwater image dehazing to aid vision-based navigation

Vidya Sudevan[1,*], **Rizwana Kausar**[1], **Sajid Javed**[1], **Hamad Karki**[1], **Giulia De Masi**[2] and **Jorge Dias**[1]

[1] Khalifa University, Abu Dhabi, United Arab Emirates
[2] Sorbonne University, Abu Dhabi, United Arab Emirates
[*] Author to whom any correspondence should be addressed.

E-mail: vidya.sudevan@ku.ac.ae

Keywords: image formation model, lightweight models, spatio-temporal features, spiking neural network, spiking transformers, underwater image dehazing

Abstract

In this paper, we introduce snnTrans-DHZ, a lightweight spiking transformer architecture that incorporates a learnable threshold membrane potential mechanism for underwater image dehazing. Despite its compact design with 0.57 M parameters, the method substantially improves underwater image clarity and visibility. Leveraging the temporal dynamics of spiking neural networks, snnTrans-DHZ efficiently processes time-dependent raw image sequences while maintaining low power consumption. The raw underwater images are first converted into time-dependent image sequences by repeatedly passing the static image to a user-defined timestep value. The RGB sequences are then converted into LAB color space representations and processed simultaneously. The architecture integrates three primary modules: (i) *K estimator* module to extract features from different color space representations, (ii) *background light estimator* module to jointly estimate the background light component from the RGB-LAB color space representations, and (iii) *soft image reconstruction* to reconstruct the haze-free, visibility-enhanced image. The snnTrans-DHZ model is directly trained using surrogate gradient-based backpropagation through time strategy. In this research, a combined loss function is designed and used. Our model is trained and tested on the UIEB and EUVP, the two publicly available benchmark dataset for image dehazing. Our algorithm achieves a PSNR of 21.6773 dB and SSIM of 0.8795 on UIEB and, on EUVP, it achieves 23.4562 dB and 0.8439. snnTrans-DHZ algorithm achieves this algorithmic performance with fewer operations (7.42 GSOPs) and lower energy consumption of 0.0151 J compared to existing state-of-the-art image enhancement methods. It provides a 3.3 × improvement in energy efficiency over the lightest state-of-the-art transformer-based method, making it suitable for underwater robotics and environmental monitoring. The source code is available at snnTrans-DHZ.

1. Introduction

Underwater image dehazing remains a critical challenge in computer vision due to poor lighting, color distortion, and water turbidity, all of which degrade image quality [1]. Methods like Shallow-UWnet [2], WaterGAN [3], U-Trans [4], and WaterFormer [5] have introduced novel architectures, including shallow ConvBlocks, probabilistic learning, GAN-based pipelines, and transformer-based modules. Although these approaches are effective, they often fall short in energy efficiency, necessitating the exploration of novel computational paradigms.

Spiking neural networks (SNNs) are energy-efficient systems for spatio-temporal processing, leveraging temporal spike timing for reduced energy consumption [6]. Efficient SNNs for image processing are developed via ANN-to-SNN conversion, using techniques like weight normalization, or direct training with surrogate gradients [7]. While recent work has shown promise in image classification, reconstruction and denoising, these methods are limited to low-resolution images and Gaussian noise

removal [8]. To address these limitations, this study explores SNN-based approaches for high-resolution underwater image dehazing.

The key contributions include:

1. **Theoretical Advancements:**
   - *Leaky integrate-and-fire (LIF) neurons with learnable threshold membrane potential ($LTMP_{LIF}$):* integration of $LTMP_{LIF}$ neuron enhances temporal dynamics and memory retention within the snnTrans-DHZ architecture. Unlike fixed-threshold LIF neurons, the $LTMP_{LIF}$ neurons adapt their threshold membrane potential dynamically during backpropagation, enabling more efficient temporal modeling.
   - *Application-specific loss function:* development of a custom loss function tailored specifically for underwater image dehazing tasks, integrating perceptual quality metrics (SSIM, contrast enhancement) with traditional pixel-wise losses. This formulation ensures balanced optimization between haze removal, structural preservation, and natural color fidelity.
2. **Practical Implementations:**
   - *Lightweight SNN-based framework (snnTrans-DHZ):* introduction of snnTrans-DHZ, an efficient transformer-integrated SNN framework for underwater image dehazing. With only 0.5670 M parameters, the model demonstrates superior efficiency and performance compared to the baseline UIE-SNN framework.
   - *Hybrid RGB-LAB processing:* implementation of hybrid RGB-LAB color space transformation, enabling separate processing of luminance and chromaticity components. This domain-aware approach significantly improves haze removal, color correction, and overall perceptual quality in reconstructed underwater images.

The proposed snnTrans-DHZ algorithm addresses key limitations in existing transformer-based [9–11] image dehazing methods by operating fully within the spiking domain, significantly enhancing energy and parameter efficiency. Unlike conventional transformer architectures that process continuous-valued signals through multiply-and-accumulate (MAC) operations, snnTrans-DHZ utilizes binary spikes (0 or 1-valued feature maps), thus replacing computationally intensive MAC operations with simpler accumulation operations. Adaptive $LTMP_{LIF}$ neurons are strategically integrated after each layer within the multi-head self-attention mechanism to maintain binary feature maps, substantially reducing the complexity associated with matrix multiplications involving real-valued feature maps. Compared specifically to the state-of-the-art UIE-SNN [12] method, snnTrans-DHZ employs a shallower architecture with a depth of only two layers and utilizes significantly fewer channels (64 versus 1024), effectively minimizing parameter count without compromising performance. These innovations collectively position snnTrans-DHZ as a more computationally efficient and energy-conscious alternative for image dehazing tasks.

2. Related works

2.1. Learning-Based dehazing methods

Li *et al* [13] introduced WaterNet, a simple learning-based UIE model with only convolutional layers. To predict the confidence maps, this model processes three derived inputs along with the original input. Feature Transformation Units are employed to minimize color distortions and artifacts caused by white balance adjustments, histogram equalization, and gamma correction techniques. The processed inputs are then refined using confidence maps to generate the final enhanced image. Shallow-UWnet [2] employs small convolutional blocks with Conv-ReLU-Dropout layers to enhance skip connections and is optimized using a combination of multiple loss functions, including mean squared error (MSE) and visual geometry group-based perceptual loss. Fu *et al* [14] introduced PUIE-Net, a probabilistic model designed to analyze and enhance degraded underwater images. It incorporates a conditional variational autoencoder with adaptive instance normalization, followed by a consensus mechanism to ensure reliable output predictions. The PDCFNet [15] method introduced pixel difference convolution (PDC) to enhance high-frequency features, improving texture and detail restoration. It employs a cross-level feature fusion module to ensure effective interaction between multi-level features, leading to enhanced image quality. Dazhao *et al* [16] developed a physics-driven framework that simultaneously trains a deep degradation model (DDM) alongside any UIE model. The DDM is responsible for estimating critical aspects of the underwater imaging process. It guarantees more accurate depth estimation and physically consistent improvement. The DDM and UIEConv models undergo joint training. UIEConv, a fully convolutional

UIE model, enhances images by incorporating both global and local features through a dual-branch design.

With the rise of GAN models, the WaterGAN [3] model was introduced to generate synthetic underwater images from corresponding terrestrial images and depth information. This model follows an unsupervised learning approach to enhance color correction for monocular underwater images. It employs a depth estimation network to reconstruct relative depth maps and a color restoration network based on a fully convolutional encoder-decoder architecture, such as SegNet. PUGAN [17] is a physical model-guided GAN architecture for UIE. It consists of a parameter estimation subnetwork that learns physical model inversion parameters by using generated color enhancement images as auxiliary information for a two-stream interaction enhancement subnetwork. Dual discriminators enforce style and content constraints, improving authenticity and visual quality. The DGD-cGAN [18] model specifically addresses underwater haze and color distortions by incorporating water-related priors in its image restoration process. It utilizes two generators: one focuses on predicting the restored scene to enhance image clarity, while the other models the underwater image formation process, employing a specialized loss function that accounts for transmission properties and veiling light effects. Finally, in LFT-DGAN [19] method, a learnable full-frequency domain transformer is integrated within a dual generative adversarial network. It introduces a reversible convolution-based image decomposition technique to separate underwater images into low, medium, and high-frequency domains for enhanced feature extraction. Additionally, it employs image channels and spatial similarity to facilitate cross-domain interaction and utilizes a dual-domain discriminator to learn both spatial and frequency characteristics, improving image restoration quality.

The Ucolor [20] method learns feature representations from various color spaces and emphasizes the most discriminative features using a channel-attention module. It incorporates domain knowledge by utilizing the reverse medium transmission map as attention weights. LANet [21] is a supervised adaptive attention network for UIE. It begins with a multiscale fusion module that combines different spatial information. It then employs a parallel attention mechanism that focuses on illuminated features and key color variations, utilizing both pixel-wise and channel-based attention. An adaptive learning framework retains shallow details while adaptively capturing important features. The training process incorporates a multinomial loss function that combines MAE loss and perceptual loss with an asynchronous training mode to enhance the performance. The Deep-WaveNet [22] model is fine-tuned with conventional pixel-wise and feature-driven cost functions. It employs a distinct receptive field structure for each image channel, driven by its wavelength, which helps in learning diverse local and global features. These features undergo refinement through a block attention mechanism. The DICAM [23] method addresses challenges associated with uneven degradation and color inconsistencies by utilizing inception modules on individual color channels to capture feature maps at various scales. These features are then processed through a channel-wise attention module (CAM) to assess the relevance of different types of degradations. Lastly, the combined feature maps undergo further refinement with CAM to enhance the image's color richness.

The U-Trans [4] method incorporates a multiscale feature fusion transformer for channel-wise processing, coupled with a transformer module designed for global spatial feature modeling. This integration strengthens the model's capability to handle color variations and spatial regions affected by substantial attenuation. UIE-Convformer [24] is a hybrid method combining CNNs and a feature fusion transformer. It employs a multiscale U-Net structure for extracting rich texture and semantic information, while the feature fusion transformer module handles global information fusion. A jump fusion connection module between the encoder and decoder fuses multiscale features through bidirectional cross-connection and weighted fusion, enriching the feature information for image reconstruction. The MFMN model [25] is a multi-scale feature modulation network designed to balance model efficiency and reconstruction performance. It employs a multi-scale spatial feature module within a visual Transformer-like block to dynamically extract spatial features and integrates a channel mixing module to enhance feature representation across channels. ICDT-XL/2 [26] is an image-conditional diffusion transformer model for UIE tasks. It replaces the conventional U-Net in a denoising diffusion probabilistic model with transformer-based architecture. By converting degraded images into a latent space, it benefits from the scalability and efficiency of transformers for improved restoration. The CFPNet [27] model is a complementary feature perception network that embeds a transformer into CNN-based UNet3+ to fuse local and global features effectively. It employs a dual encoder structure combining CNN and transformer branches. It includes a full-scale feature fusion module for multi-scale information merging and an auxiliary feature-guided learning strategy to enhance color correction and deblurring. The MMIETransformer [10] model is a transformer-based architecture for general image enhancement, incorporating space-aware deformable convolution and multi-head self-attention for fine texture reconstruction. It consists of a spatially attentive offset extractor, an edge-enhancing feature fusion block, and

a global context-aware channel attention module to refine edge details and enhance multi-stream feature learning.

The WPFNet [28] model, introduced by Shiben *et al* utilizes a hybrid wavelet-pixel domain fusion approach to improve illumination, reduce color deviation, and improve details in degraded underwater images. It integrates a wavelet domain module to capture frequency-based multi-scale information, a residual attention block to refine low-frequency details, and a Transformer Block (T_Block) for high-frequency feature processing. Additionally, pixel domain module is developed to identify and highlight detailed spatial attributes. The DDFormer [11] method integrates a dimension decomposition transformer with semi-supervised learning for UIE. It introduces dimension decomposition attention to compute global dependencies at the original scale and correct color distortions. A multistage transformer strategy is employed in DDFormer to capture multi-scale global information. The Ghost-Unet [9] model introduced by Lingyu *et al* is an efficient UIE framework leveraging a conditional diffusion approach to reduce redundant feature maps using linear transformations. It consists of a forward diffusion process, a reverse denoising process, and a noise prediction network, aiming to generate realistic underwater degradation images from high-quality inputs. Zhiqiang *et al* proposed the Dynamic SpectraFormer [29] method to enhance underwater images using a frequency domain transformer combined with an ultra-high-resolution sparse spectrum attention module for long-term dependency modeling. It employs a dynamic spectrum weight generation layer that acts as an adaptive spectrum band selector, emphasizing critical frequency bands while suppressing irrelevant ones for improved enhancement. Transformer-based models demonstrate performance comparable to CNNs by effectively capturing long-range relationships and expanding receptive areas across various visual tasks. However, their application and deployment are challenged by the large number of parameters involved.

2.2. SNN-based pixel-prediction methods

Developing deep SNNs for visual data processing can be achieved through two primary strategies [30]. The first strategy involves transforming conventional artificial neural networks (ANNs) into their SNN equivalents, whereas the second focuses on training SNNs from scratch using spike-based learning techniques. In the ANN-to-SNN conversion approach, pre-trained ANN models are adapted into spiking architectures. To regulate spike activity in classification tasks, Diehl *et al* introduce a layer-wise weight normalization technique in combination with threshold tuning [31]. Additionally, Yan *et al* present a clamped and quantized training approach designed to reduce information loss during the transformation, ensuring minimal degradation in classification accuracy [32].

The direct training of SNN can be accomplished through supervised or unsupervised learning paradigms. Within the scope of unsupervised learning, the spike-timing-dependent plasticity (STDP) strategy is recognized as a foundational approach. The method has been applied in various tasks, such as training networks for image classification [33] and object recognition using event-driven data representations [34]. Additionally, STDP has been integrated with dynamic threshold adjustments to enhance recognition capabilities in video-based facial disguise detection [35]. Supervised learning in SNNs presents unique challenges due to the discrete nature of spike-based neuron activations, making direct gradient computation difficult. To address this, [36] introduces an approach that smooths spike events, enabling gradient propagation across network layers using chain rule-based computations. A more structured approach, spatial-temporal backpropagation (STBP), is leveraged for image classification tasks by incorporating surrogate gradients to approximate non-differentiable spike-based updates [37]. A recent advancement explores a hybrid learning framework that merges STDP with STBP to enhance object detection capabilities in SNNs [38]. This integration aims to leverage the efficiency of unsupervised feature extraction while benefiting from the structured optimization of supervised learning techniques. The summary of different SNN training strategies is detailed in table 1.

Roy *et al* [39] introduced a method to synthesize images from multiple modalities within a spike-based representation. They utilized a spiking autoencoder to encode inputs into compact spatio-temporal representations, which were then decoded for image reconstruction. The training process involved computing the membrane potential loss at the output layer and backpropagating it using a sigmoid approximation of the neuron's activation function to ensure differentiability. Later, in 2021, a spiking autoencoder with temporal coding and pulse-based training was presented [40], demonstrating the importance of inhibition for memorizing inputs over extended periods before generating the expected output spikes.

Castagnetti *et al* [8] introduced the first SNN-driven Gaussian image denoising model, employing approximate gradient optimization and temporal BPTT techniques to train the network within the spiking framework. In 2024, two advancements were introduced. A spiking U-Net using multi-threshold neurons was proposed, leveraging an ANN-SNN adaptation framework followed by refinement using previously trained U-Net networks [7]. Additionally, a non-training-based method was proposed,

Table 1. Comparison between different SNN training strategies.

Training strategy	Key methods	Advantages	Disadvantages	Use cases
ANN to SNN Conversion	Spike coding; layer-wise normalization	Utilizes pre-trained ANN models; Works well for deep networks	High latency; Inefficient for real-time applications	Deep learning tasks
Spike-based backpropagation	Surrogate gradient descent; back propagation through time (BPTT)	Enables direct SNN training; Achieves high accuracy	Computationally expensive; memory-intensive	Supervised learning; Large-scale deep SNNs
Local learning rules	Spike-timing-dependent plasticity(STDP); Hebbian learning	Hardware-efficient; suitable for neuromorphic hardware	Difficult to achieve high accuracy; No direct supervision	Unsupervised learning; Low-power edge computing on neuromorphic chips
Evolutionary & reinforcement learning	Evolutionary Strategies(ES); reinforcement learning(RL); Neuromodulated Plasticity	suitable for decision-making	Slow convergence; high variance	Robotics; Adaptive control; Decision-making tasks
Hybrid learning	STDP+Back Propagation; ANN-SNN Hybrid Training; Meta-learning	Balances computational efficiency and accuracy; Adaptable to different applications	Increased complexity; requires fine-tuning	Optimization across multiple tasks

utilizing threshold guidance for SNN-based diffusion models in Gaussian image denoising [41]. Additionally, ESDNet [42], an image-deraining model, introduced a spiking residual block that converted inputs into spike signals, optimizing membrane potentials adaptively with attention weights to mitigate information loss from discrete binary activations. While these methods have shown promise in image reconstruction and denoising, they are limited to low-resolution images and Gaussian noise removal. This makes them inadequate to use for high-resolution, complex underwater image sequences with non-linear haze distributions.

Despite the progress of direct SNN training and recent SNN-based image restoration models, most existing approaches are primarily evaluated on low-resolution inputs and relatively constrained degradations. These methods do not address the coupled challenges of wavelength-dependent attenuation, severe color cast, and spatially varying backscatter that characterize underwater haze. In contrast, our proposed approach targets high-resolution underwater image dehazing and introduces several design choices that directly respond to these gaps. First, we incorporate LTMP_{LIF} neurons with learnable threshold membrane potential, enabling each spiking layer to adapt its firing rate during network training. Second, we integrate a lightweight spiking transformer module to capture long-range spatial dependencies directly in the spike-based feature maps. This design avoids conventional softmax-normalized self-attention and instead employs a spiking, softmax-free attention mechanism, which reduces computational overhead in SNN implementations. Third, we adopt a hybrid RGB-LAB processing strategy that separately leverages luminance and chromaticity cues. This is particularly important for underwater scenes where color distortions and illumination variation are dominant failure modes. Finally, we employ an application-specific loss function that balances pixel fidelity, structural similarity, and smoothness priors to improve perceptual quality and color fidelity in the reconstructed output. These components collectively differentiate snnTrans-DHZ from existing SNN-based pixel-prediction methods and enable effective underwater dehazing with reduced model complexity and energy consumption.

2.2.1. Conversion of continuous pixel intensities to spikes

Converting continuous-valued input signals to a sparse, spike-based representation without significant information loss is one of the most important tasks in any SNN-based computing paradigm. Several spike-coding-decoding techniques have been proposed to tackle the current lack of neuromorphic sensing for robotics [43, 44]. Temporal coding [45] uses spike timing that is inversely proportional to pixel

intensity. The neuron that receives the highest activation value fires a spike first compared to the neuron that receives a lower value. Since there is only one spike firing at each timestep, the average spike rate remains lower, requiring larger timesteps to capture relevant information [46].

In phase coding, temporal information is encoded into spike representations using a global oscillator [47]. In binary phase coding, a different weight determined by the power of 2 is assigned to each timestep. Thus, the activation values are encoded by both the spike pattern and spike rates. These methods are not suited for architectures where network size and dataset size are large [48]. Rate coding [49] is the widely used spike-coding technique in classification tasks, where the input value is encoded into a spike train over a pre-defined number of timesteps. The number of generated spikes, sampled from the Poisson distribution, is proportional to the input magnitude. In the direct coding technique [50], the continuous-valued input is fed directly to the first learning-based layer of the network, which outputs the weighted float-point values. The subsequent spiking neuron layer repeatedly receives these values over the pre-defined timesteps to generate spikes.

3. Knowledge background

3.1. Spiking neuron with adaptive threshold

The LIF neuron dynamics with a modified adaptive threshold mechanism are used in the spike coding module along with the convolutional layers. The neuron dynamics of an LIF neuron are modeled using an *RC* circuit [51] and is represented as:

$$\tau \frac{\mathrm{d}V(t)}{\mathrm{d}t} = -V(t) + I(t)R. \tag{1}$$

Here, τ represents the time constant ($\tau = RC$), with R and C denoting the resistive and capacitive components of the *RC* circuit, respectively. The function $I(t)$ corresponds to the applied input current. For a constant input $I(t) = I$, the solution to (1) is:

$$V(t) = IR + [V_0 - IR]\,\mathrm{e}^{-t/\tau}. \tag{2}$$

To make this model compatible with sequence-based neural networks, we discretize time $t \to t+1$ with a small step Δt. Applying the Euler method to (1):

$$\frac{V(t+\Delta t) - V(t)}{\Delta t} = -\frac{V(t)}{\tau} + \frac{IR}{\tau}. \tag{3}$$

Rearranging (3) for $V(t+\Delta t)$:

$$V(t+\Delta t) = V(t) + \frac{\Delta t}{\tau}\left(-V(t) + IR\right). \tag{4}$$

Assuming $\Delta t = 1$, we define the decay rate $\zeta = \mathrm{e}^{-1/\tau}$. Thus, the update equation simplifies to:

$$V[t] = \zeta V[t-1] + (1-\zeta)\,I[t]. \tag{5}$$

The above formula is the Euler-approximated discretized form of the LIF dynamic equation. In the learning-based frameworks, the weighting factor $(1-\zeta)$ of the input $I[t]$ is considered as a learnable parameter. Instead of directly using $I[t]$, we define $I[t] = ZX[t]$. $X[t]$ represents the neuron's received signal, while Z denotes the adaptive synaptic weight acquired through learning. Therefore,

$$V[t] = \zeta V[t-1] + ZX[t]. \tag{6}$$

Neurons exhibit a spiking behavior. That is, if the internal potential V exceeds a defined threshold V_{th}, it triggers a spike firing, followed by a reset of the neuron's state. The spiking event follows the formulation $S[t] = \Theta(V[t] - V_{\mathrm{th}})$. Here $\Theta(V[t] - V_{\mathrm{th}})$ is the Heaviside function, which is defined as:

$$\Theta\left(V[t] - V_{\mathrm{th}}\right) = \begin{cases} 1, & V[t] \geqslant V_{\mathrm{th}}. \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

To enforce the reset after a spike, we subtract the threshold potential weighted by the spike output:

$$V[t] = \zeta V[t-1] + ZX[t] - S[t-1]\,V_{\mathrm{th}}. \tag{8}$$

In the direct coding scheme with an adaptive threshold, the threshold membrane potential V_{th}, is considered as learnable parameter.

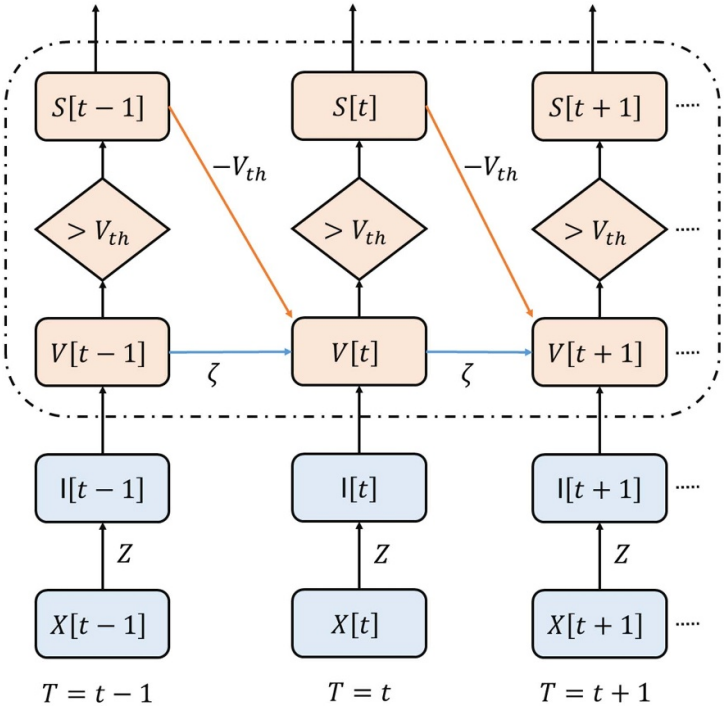

Figure 1. Computational graph of LIF neuron over the timesteps.

3.2. Training strategies

Unlike the widely adopted gradient descent optimization used for training conventional ANNs [52], optimizing SNNs presents significant challenges because spike-based computations are inherently non-differentiable. The three main approaches are: (i) conversion from ANN to SNN [53, 54], where a conversion from formal to event-driven input can lead to a large loss of accuracy and poor interpretability; ii) supervised learning, in which various approaches aim to address the fundamental limitation of spike-based mathematical expressions-specifically, their non-differentiability, which is essential for any back-propagation algorithm. iii) unsupervised methods, such as STDP, in which synaptic weight adjustments are influenced by the temporal relationship of spike events from pre- and post-synaptic neurons. (If the timing is short, this implies causality and large weight and vice versa). The training procedures that are closely related to its biological counterpart are *local training* and *reinforcement learning* [55]. The local training strategy is characterized by high *adaptability* to the changes of inputs and environments, quite differently from traditional neural networks.

In reference to figure 1, the backpropagation algorithm in SNN-based model at a single timestep is formulated,

$$\frac{\partial \mathcal{L}}{\partial Z} = \frac{\partial \mathcal{L}}{\partial S}\frac{\partial S}{\partial V}\frac{\partial V}{\partial I}\frac{\partial I}{\partial Z}. \tag{9}$$

The output spike generated during the forward propagation is given as:

$$S[t] = \Theta\left(V[t] - V_{\text{th}}\right) = \begin{cases} 1, & \text{if } V \geqslant V_{\text{th}}, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The derivative calculated during the backward propagation for (10) is:

$$\frac{\partial S}{\partial V[t]} = \delta\left(V - V_{\text{th}}\right) = \begin{cases} +\infty, & \text{if } V[t] = V_{\text{th}}, \\ 0, & \text{if } V[t] \neq V_{\text{th}}, \end{cases} \tag{11}$$

where, $\delta(.)$ is the Dirac-Delta function.

This implies that the term $\frac{\partial S}{\partial V}$ will almost always be zero, thereby inhibiting the learning process. In this case, the network training will be unstable with the use of $\delta(.)$ function to calculate the gradient and apply the gradient descent.

The surrogate gradient-based method with fast-sigmoid function is used in this work to mitigate the dead neuron problem [51]. The threshold-centered membrane potential is defined as $\tilde{V} = V[t] - V_{\text{th}}$. The fast-sigmoid function and its derivative used for backpropagation is represented as:

$$\tilde{S} \approx \frac{\tilde{V}}{\left(1 + \lambda \left|\tilde{V}\right|\right)}, \tag{12}$$

$$\frac{\partial \tilde{S}}{\partial V} \approx \frac{1}{\left(1 + \lambda \left|\tilde{V}\right|\right)^2}, \tag{13}$$

where λ modulates the smoothness of the surrogate function.

Since SNNs operate in the temporal domain, the weight Z influences the membrane potential and loss across all time steps. Since the weights are shared across all timesteps (Z remains unchanged for $Z[0], Z[1], \ldots, Z[T]$), the global loss gradient $\mathcal{L}$ computed over the weight Z for all past timesteps $p \leqslant t$ is defined as:

$$\frac{\partial \mathcal{L}}{\partial Z} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}[t]}{\partial Z} = \sum_{t=1}^{T} \sum_{p \leqslant t} \frac{\partial \mathcal{L}[t]}{\partial Z[p]} \frac{\partial Z[p]}{\partial Z} = \sum_{t=1}^{T} \sum_{p \leqslant t} \frac{\partial \mathcal{L}[t]}{\partial Z[p]}. \tag{14}$$

For instance, if only the immediate prior influence corresponding to timestep $p = t - 1$ is taken into account, the backward pass needs to be traced back in time by a single step. In this case, the influence of $Z[t-1]$ on $\mathcal{L}[t]$ is described as:

$$\frac{\partial \mathcal{L}[t]}{\partial Z[t-1]} = \frac{\partial \mathcal{L}[t]}{\partial S[t]} \frac{\partial \tilde{S}[t]}{\partial V[t]} \frac{\partial V[t]}{\partial V[t-1]} \frac{\partial V[t-1]}{\partial I[t-1]} \frac{\partial I[t-1]}{\partial Z[t-1]}. \tag{15}$$

Here, $\frac{\partial V[t]}{\partial V[t-1]}$ denotes the decay value defined for a LIF neuron; $\frac{\partial V[t-1]}{\partial I[t-1]} = 1$, and $\frac{\partial I[t-1]}{\partial Z[t-1]}$ is the pre-synaptic input.

By combining surrogate gradients with BPTT, the training of SNNs becomes both feasible and efficient. The surrogate gradient approximations enable non-zero gradients even if the neuron's internal voltage approaches its activation limit. The BPTT strategy ensures that the temporal dependencies are captured across timesteps. The surrogate gradient used in this work aligns with established methods in the literature [51], and provides smooth gradient flow while preserving the temporal dynamics essential for SNNs.

3.3. Spike coding of continuous-values signals

A LIF neuron with an adaptive threshold mechanism is used after the convolutional input layer for the spike coding of continuous input values. For processing the visual inputs, a 2D convolutional layer followed by the adaptive LIF neuron converts the continuous pixel intensity values to their equivalent binary spike representation. In the decoding phase, these spikes are then fed to the output convolutional layer followed by a LIF neuron whose membrane potential, without resetting, is used directly to compute the reconstruction loss. The entire spike-coding process of a single image during a single timestep is presented in figure 2.

3.4. Energy measurement

The energy consumption for CNNs and SNNs are computed based on the total number of floating point operations (FLOPs) and synaptic operators (SOPs) respectively. The energy consumption is based on the standard CMOS technology [56] as shown in table 2.

The FLOPs and SOPs for each convolution layer l in CNN and SNN are computed as follows:

$$\begin{aligned} CNN: \quad & FLOPs_{\text{CNN}}(l) = l_{\text{Cin}} \quad l_{\text{Cout}} \quad l_k^2 \quad l_l \quad l_w \\ SNN: \quad & SOPs_{\text{SNN}}(l) = FLOPs_{\text{CNN}}(l) \quad S_r(l) \end{aligned} \tag{16}$$

where, l_{Cin} is the number of input channels at layer l, l_{Cout} is the number of output channels at layer l, l_k is the size of the kernel at layer l, l_l is the length of the output feature map at layer l, l_w is the width of the output feature map at layer l and $S_r(l)$ is the spike rate [57] at each SNN layer l.

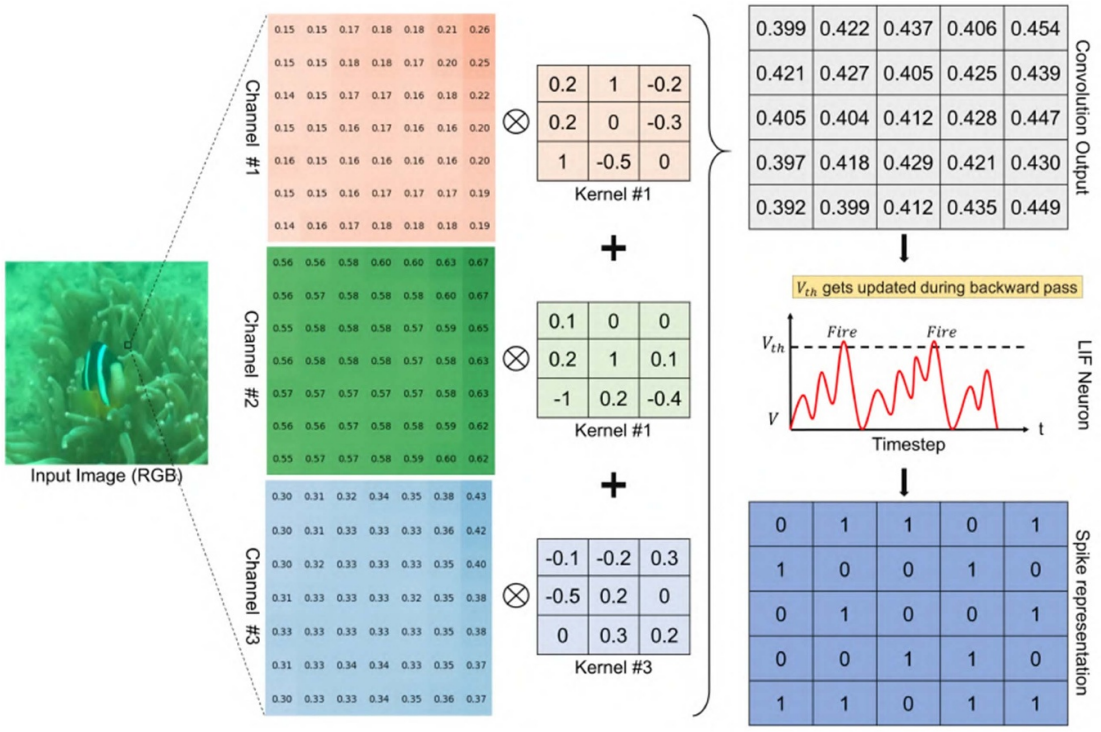

Figure 2. Spike coding using LIF neuron with adaptive threshold: In the first stage, feature maps are generated when each channel of the input RGB images, and undergoes convolution with respective kernel. The output is then fed to a spiking neuron, where the membrane potential (V) is updated depending on the presynaptic input. When V exceeds the threshold membrane potential (V_{th}), a spike is fired, and V resets to a predefined value. In the adaptive threshold mechanism, V_{th} is updated during backpropagation,

Table 2. Energy consumption for 45 nm CMOS process.

Floating Point (FP) Operation	Energy (pJ)
32 bit FP MULT	3.7
32 bit FP ADD	0.9
32 bit FP MAC (E_{MAC})	4.6
32 bit FP ACC (E_{ACC})	0.9

$$S_r(l) = \frac{\textit{Number of spikes over all timesteps}}{\textit{Number of neuron}} = \frac{\sum_{c=1}^{l_{\text{Cout}}} \sum_{i=1}^{l_l} \sum_{j=1}^{l_w} \sum_{t=1}^{T} S_l^t(i,j,c)}{l_{\text{Cout}} \quad l_l \quad l_w} \tag{17}$$

The spike rate $S_r(l)$ effectively scales the computational cost of SNNs by accounting for spike-driven sparse computations. Since SNNs operate on binary spikes rather than continuous activations, the overall number of operations can be significantly reduced compared to traditional CNNs.

Given the operation counts per layer, the total inference energy for CNNs and SNNs across all layers can be determined as follows:

$$\begin{aligned} \text{CNN} &: E_{\text{CNN}} = E_{\text{MAC}} \left(\sum_{l=1}^{L} \text{FLOPs}_{\text{CNN}}(l) \right) \\ \text{SNN} &: E_{\text{SNN}} = E_{\text{ACC}} \left(\sum_{l=1}^{L} \text{SOPs}_{\text{SNN}}(l) \right) \end{aligned} \tag{18}$$

where L is the total number of layers in the architecture, E_{MAC} represents the energy per MAC operation, and E_{ACC} represents the energy per accumulation operation. E_{MAC} and E_{ACC} can be obtained from table 2.

The percentage of energy reduction is calculated as:

$$\Delta E(\%) = 100\left(\frac{E_{\text{CNN}} - E_{\text{SNN}}}{E_{\text{CNN}}}\right) \tag{19}$$

4. Modelling of snnTrans-DHZ architecture

The schematic representation of the proposed snnTrans-DHZ framework for underwater image dehazing is presented in figure 3. The raw underwater images are first converted into time-dependent image sequences by repeatedly passing the static image to a user-defined timestep value T. The RGB sequences are then converted into LAB color space representations and processed simultaneously. The *K estimator* utilizes spiking transformer-based architecture along with convolutional SNN layers to extract features from different color space representations. A joint feature upsampling is performed using the deconvolution layers, which leads to the extraction of **K**. The time-dependent image sequences in RGB and LAB color spaces are concatenated and fed to the convolutional SNN layers to jointly estimate the background light estimate **B**. Finally, the *soft image reconstruction* module reconstructs the haze-free, visibility-enhanced image $\hat{\mathbf{Y}}$ by utilizing **K**, **B** and $\mathbf{X}_{\text{img}}^{\text{RGB}}$.

4.1. LIF neurons with learnable threshold membrane potential (LTMP_{LIF})

The LTMP_{LIF} neuron modifies the standard LIF model by replacing the fixed threshold with a learnable threshold V_{th}. It allows the LTMP_{LIF} neuron to dynamically adjust its firing behavior based on past activity. Unlike the LIF neuron with a fixed threshold membrane potential, in LTMP_{LIF} neurons, the V_{th} is treated as a learnable parameter optimized through gradient descent. This adaptation eliminates the need for tedious selection of an optimal V_{th}. By making V_{th} adaptive, the LTMP_{LIF} neuron enhances temporal coding and learning capabilities, making it more biologically plausible and efficient for SNN-based applications.

In snnTrans-DHZ, the learnable threshold is defined in a layer-wise manner. Specifically, each LTMP_{LIF} layer maintains its own learnable threshold parameter V_{th}, and these thresholds are not shared across layers. Within a given layer, the V_{th} is parameterized as a single scalar and is therefore shared across all neurons (i.e. across channels and spatial locations) governed by that layer. During training, V_{th} is optimized jointly with the synaptic weights via surrogate-gradient backpropagation-through-time, driven by the network objective $\mathcal{L}_{\text{net}}$ (refer to section 4.4). The gradient-based update is $V_{\text{th}} \leftarrow V_{\text{th}} - \eta \frac{\partial \mathcal{L}_{net}}{\partial V_{\text{th}}}$, where η denotes the learning rate. This formulation allows each layer to autonomously regulate the spike firing during training.

4.2. RGB to LAB transformation

The LAB representation offers a perceptually consistent color representation, closely aligning with human visual perception to ensure accurate color differentiation. It consists of three components: L (lightness), A (green-red), and B (blue–yellow), providing a device-independent representation of color. This conversion enhances robustness in various computer vision tasks by reducing sensitivity to illumination changes. The conversion from RGB to LAB is performed through an intermediate *XYZ* color space. Given an RGB image sequence $\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}}$, each pixel's linear RGB values (R, G, B) are first normalized to $[0, 1]$ and then converted to CIEXYZ using the standard sRGB transformation matrix.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{20}$$

For gamma correction, $\{R, G, B\}$ values are first linearized [58] using:

$$C_{\text{linear}} = \begin{cases} \frac{C}{12.92} & , C \leqslant 0.04045 \\ \left(\frac{C+0.055}{1.055}\right)^{2.4} & , C > 0.04045 \end{cases} \tag{21}$$

where $C \in \{R, G, B\}$. The $\{X, Y, Z\}$ values are mapped to LAB color space using the following nonlinear transformation:

$$f(u) = \begin{cases} u^{\frac{1}{3}} & , u > 0.008856 \\ 7.787u + \frac{4}{29} & , u \leqslant 0.008856 \end{cases} \tag{22}$$

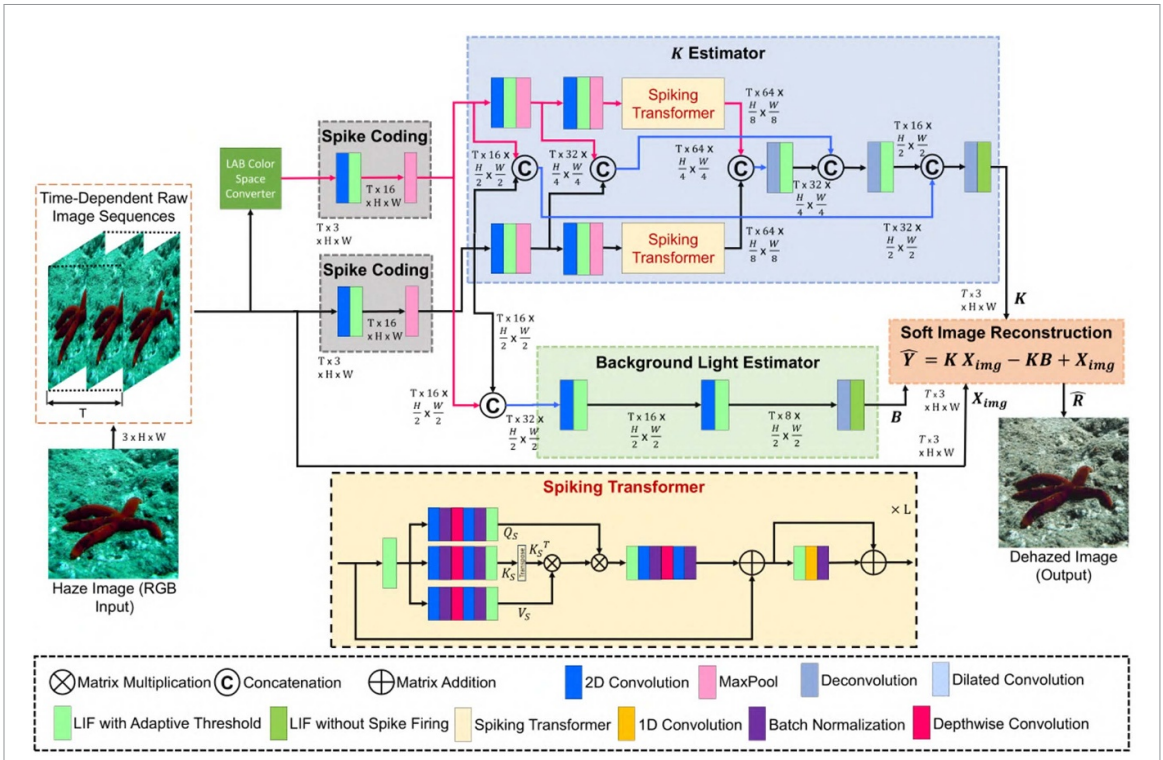

Figure 3. Schematic representation of snnTrans-DHZ framework for underwater image dehazing. The spike coding is performed simultaneously on the RGB and LAB color space converted time-dependent raw image sequences. In the *K estimator* module, the low-level features from RGB and LAB color spaces are extracted separately using the spike transformer. They are concatenated and fed to the deconvolution layer. The separately extracted feature maps at different spatial resolutions are fused and fed to the corresponding deconvolution layers. In the *background light estimator* module, RGB-LAB spike-coded signals are concatenated and fed to the convolution layers, followed by LTMP_{LIF} neurons for feature extraction. The **K** and **B** estimates, together with $\mathbf{X}_{\text{img}}^{\text{RGB}}$, are fed to the *soft image reconstruction* module for the reconstruction of haze-free images. The LTMP_{LIF} neurons are used throughout the network.

where, u is $\frac{X}{X_n}$, $\frac{Y}{Y_n}$, or $\frac{Z}{Z_n}$. The LAB components are then computed as:

$$\begin{aligned} L &= \quad 116f\left(\frac{Y}{Y_n}\right) - 16 \\ A &= 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ B &= 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{aligned} \tag{23}$$

where, (X_n, Y_n, Z_n) represent the white achromatic reference illuminant. Since the input image is time-dependent, we apply the conversion frame-wise to the image sequence:

$$\tilde{\mathbf{X}}_{\text{img}}^{\text{LAB}} = \text{RGB} \rightarrow \text{LAB}\left(\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}}\right) \tag{24}$$

where $\tilde{\mathbf{X}}_{\text{img}}^{\text{LAB}}$ represents the LAB image sequences over T timesteps.

4.3. Model description

The raw RGB input image $\mathbf{X}_{\text{img}}^{\text{RGB}} \in \mathbb{R}^{3\times H\times W}$, is first converted to time-dependent image sequences $\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}} \in \mathbb{R}^{T\times 3\times H\times W}$. The time-dependent sequence $\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}}$ is formed by directly repeating the same continuous-valued static image across the user-defined timestep length T. For each timestep $t \in \{1, 2, .., T\}$, the sequence is defined as $\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}}[t] = \mathbf{X}_{\text{img}}^{\text{RGB}}$, such that every frame in the temporal sequence is identical to the original input image. No additional perturbation is applied during this conversion. Consequently, the temporal dynamics exploited by snnTrans-DHZ arise from the internal state evolution of the spiking neurons (membrane potential accumulation, leakage, and threshold-based firing) when processing the repeated input over T timesteps. These dynamics are therefore induced by the network's spiking mechanisms rather than by any externally simulated temporal variations. RGB image sequences $\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}}$ are then converted to $\tilde{\mathbf{X}}_{\text{img}}^{\text{LAB}} \in \mathbb{R}^{T\times 3\times H\times W}$. The continuous-valued time-dependent input image

sequences $\tilde{\mathbf{X}}_{\text{img}}^{\text{RGB}}$ and $\tilde{\mathbf{X}}_{\text{img}}^{\text{LAB}}$ are processed through two distinct spike coding modules, transforming them into spike-based representations. The spike-coded RGB and LAB branch outputs after downsampling are given by $\mathbf{S}_{\text{RGB}} \in \{0,1\}^{T\times 16\times \frac{H}{2}\times \frac{W}{2}}$ and $\mathbf{S}_{\text{LAB}} \in \{0,1\}^{T\times 16\times \frac{H}{2}\times \frac{W}{2}}$.

4.3.1. Background light estimator

This module estimates the background light $\mathbf{B}$ from the concatenated spike-coded features of the RGB and LAB pathways. For the concatenated RGB and LAB spike-coded outputs along the channel dimension, be $\mathbf{X}_{\text{BL}} = \text{concat}(\mathbf{S}_{\text{RGB}}, \mathbf{S}_{\text{LAB}}) \in \{0,1\}^{T\times 32\times \frac{H}{2}\times \frac{W}{2}}$. Upon sequential convolution and deconvolution layers followed by the LTMP_{LIF} neuron, the $\mathbf{B}$ is obtained by:

$$\mathbf{B} = \text{LIF}_{\text{final}}(\text{DeConv}(\text{LTMP}_{\text{LIF}}(\text{Conv}(\text{LTMP}_{\text{LIF}}(\text{Conv}(\mathbf{X}_{\text{BL}})))))) \in \mathbb{R}^{T\times 3\times H\times W}. \tag{25}$$

4.3.2. K estimator

The goal of this module is to estimate the spatially varying transmission function $\mathbf{K} \in \mathbb{R}^{T\times 3\times H\times W}$. It comprises dual encoder streams, with one handling RGB spike-coded features while the other processes LAB spike-coded features and a single decoder that upsamples the combined features to produce the final $\mathbf{K}$ estimate.

There are two encoding stages for the RGB and LAB branches, followed by a spiking transformer module. Each encoding stage consists of a convolution layer, followed by a LTMP_{LIF} neuron layer and a downsampling block to reduce the spatial dimension.

Encoder stage 1:

$$\mathbf{S}_{\text{RGB},e1}^{K} = \text{MaxPool}(\text{LTMP}_{\text{LIF}}(\text{Conv}(\mathbf{S}_{\text{RGB}}))) \in \{0,1\}^{T\times 32\times \frac{H}{4}\times \frac{W}{4}}, \tag{26}$$

$$\mathbf{S}_{\text{LAB},e1}^{K} = \text{MaxPool}(\text{LTMP}_{\text{LIF}}(\text{Conv}(\mathbf{S}_{\text{LAB}}))) \in \{0,1\}^{T\times 32\times \frac{H}{4}\times \frac{W}{4}}, \tag{27}$$

Encoder stage 2:

$$\mathbf{S}_{\text{RGB},e2}^{K} = \text{MaxPool}(\text{LTMP}_{\text{LIF}}(\text{Conv}(\mathbf{S}_{\text{RGB},e1}^{K}))) \in \{0,1\}^{T\times 64\times \frac{H}{8}\times \frac{W}{8}}, \tag{28}$$

$$\mathbf{S}_{\text{LAB},e2}^{K} = \text{MaxPool}(\text{LTMP}_{\text{LIF}}(\text{Conv}(\mathbf{S}_{\text{LAB},e1}^{K}))) \in \{0,1\}^{T\times 64\times \frac{H}{8}\times \frac{W}{8}}. \tag{29}$$

4.3.2.1. Spiking Transformer Blocks

A series of transformer blocks is applied to the downsampled features for each branch. One transformer block $\mathcal{B}$ comprises two main parts: a spiking self-attention module and an MLP block with residual connections. Unlike the Vanilla self-attention (VSA) [59], a group of convolution and batch normalization operations is performed to extract spike-based query ($\mathbf{Q}_S$), key ($\mathbf{K}_S$), and value ($\mathbf{V}_S$) matrix from the input of the transformer block. The key differences of VSA and adaptive spike-based self-attention (ASBSA) is presented in figure 4. The group operations in the transformer module, G^{Trans}, consist of a convolution layer, followed by batch normalization, depth-wise convolution layer, convolution layer, and batch normalization. The convolution layer defined here is two-dimensional. Otherwise, it will be explicitly mentioned.

For the extraction of spike-based query ($\mathbf{Q}_S$), key ($\mathbf{K}_S$), and value ($\mathbf{V}_S$) for a given input $\mathbf{X}_{\text{in}}^{\text{Trans}} \in \mathbb{R}^{T\times 64\times \frac{H}{8}\times \frac{W}{8}}$:

$$\mathbf{Q}_S = \text{LTMP}_{LIF}(G^{\text{Trans}}(\text{LTMP}_{\text{LIF}}(\mathbf{X}_{\text{in}}^{\text{Trans}})) \in \{0,1\}^{T\times N\times D}, \tag{30}$$

$$\mathbf{K}_S = \text{LTMP}_{\text{LIF}}(\text{G}^{\text{Trans}}(\text{LTMP}_{\text{LIF}}(\mathbf{X}_{\text{in}}^{\text{Trans}})) \in \{0,1\}^{T\times N\times D}, \tag{31}$$

$$\mathbf{V}_S = \text{LTMP}_{\text{LIF}}(\text{G}^{\text{Trans}}(\text{LTMP}_{\text{LIF}}(\mathbf{X}_{\text{in}}^{\text{Trans}})) \in \{0,1\}^{T\times N\times D}, \tag{32}$$

where N is the token number, and D represents the dimensionality of embedded vector.

The interaction between $\mathbf{Q}_S, \mathbf{K}_S, \mathbf{V}_S$ within the ASBSA is formulated as:

$$\text{ASBSA}(\mathbf{Q}_S, \mathbf{K}_S, \mathbf{V}_S) = \text{LTMP}_{\text{LIF}}\left(\mathbf{Q}_S\left(\mathbf{K}_S^T\mathbf{V}_S\right)\right) \in \{0,1\}^{T\times N\times D}. \tag{33}$$

The input $\mathbf{X}_{\text{mlp}}^{\text{Trans}}$ to the transformer linear layer is:

$$\mathbf{X}_{mlp}^{\text{Trans}} = G^{Trans}(ASBSA(\mathbf{Q}_S, \mathbf{K}_S, V_S)) + \mathbf{X}_{in}^{\text{Trans}} \in \mathbb{R}^{T\times 64\times \frac{H}{8}\times \frac{W}{8}}. \tag{34}$$

The output of the spike-based T_Block is:

$$\mathbf{X}_{out}^{\text{Trans}} = \mathbf{X}_{mlp}^{\text{Trans}} + \text{BN}(\text{Conv1d}\left(\text{LTMP}_{LIF}\left(\mathbf{X}_{mlp}^{\text{Trans}}\right)\right) \in \mathbb{R}^{T\times 64\times \frac{H}{8}\times \frac{W}{8}}. \tag{35}$$

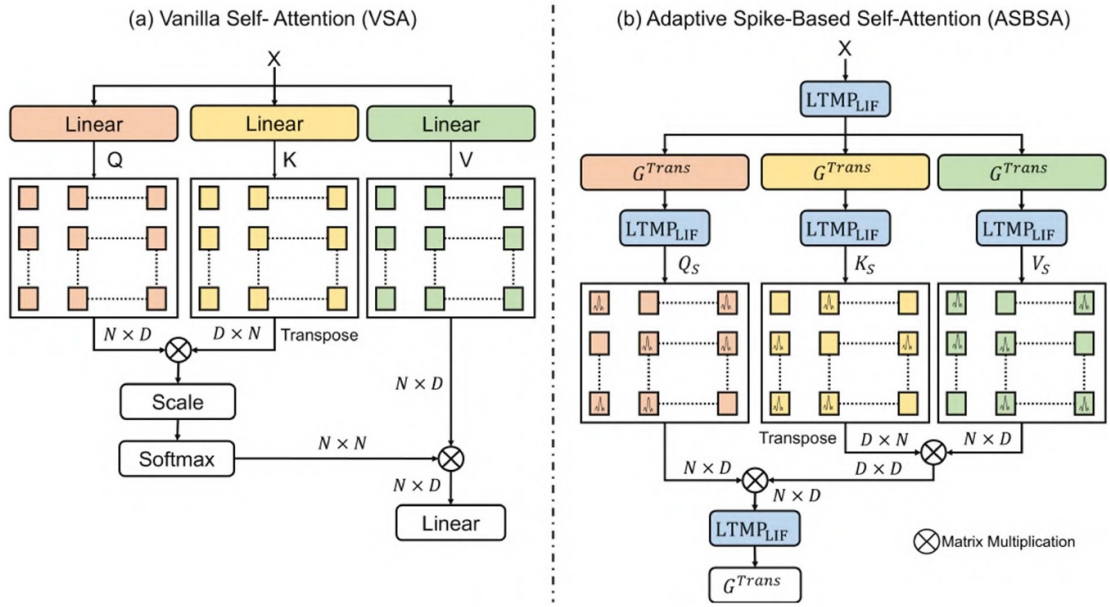

Figure 4. Key differences of Vanilla self-attention (VSA) and adaptive spike-based self-attention (ASBSA). (a) VSA follows the traditional attention mechanism, where input X is mapped into query (Q), key (K), and value (V) representations, followed by scaled dot-product attention and softmax normalization; (b) SBSA introduces LTMP_{LIF} neurons with a learnable threshold potential and spike-based transformation module G^{Trans} replacing traditional linear layers. This enhances energy efficiency and sparsity by leveraging spiking neural dynamics. The attention computation retains matrix multiplications but eliminates the scale and softmax operations.

Considering that the RGB and LAB pathway inputs to the spiking transformer modules are $\mathbf{S}^K_{\text{RGB},e2}$ and $\mathbf{S}^K_{\text{LAB},e2}$ generate the output as $\mathbf{S}^K_{\text{RGB,trans}}$ and, $\mathbf{S}^K_{\text{LAB,trans}}$ respectively. The feature maps at the corresponding spatial resolution are concatenated and fed to the decoder layers, which finally outputs $\mathbf{K} \in \mathbb{R}^{T\times 3\times H\times W}$.

Output at decoder stage 1:

$$\mathbf{S}^K_{d1} = \text{LIF}(\text{DeConv}(\text{concat}(\mathbf{S}^K_{\text{RGB,trans}}, \mathbf{S}^K_{\text{LAB,trans}}))) \in \{0,1\}^{T\times 32\times \frac{H}{4}\times \frac{W}{4}}. \tag{36}$$

Output at decoder stage 2:

$$\mathbf{S}^K_{d2} = \text{LTMP}_{\text{LIF}}(\text{DeConv}(\text{concat}(\mathbf{S}^K_{d1}, \text{concat}(\mathbf{S}^K_{\text{RGB},e1}, \mathbf{S}^K_{\text{LAB},e1})))) \in \{0,1\}^{T\times 16\times \frac{H}{2}\times \frac{W}{2}}. \tag{37}$$

Output at decoder stage 3:

$$\mathbf{K} = \text{LIF}_{\text{final}}(\text{DeConv}(\text{concat}(\mathbf{S}^K_{d2}, \text{concat}(\mathbf{S}_{\text{RGB}}, \mathbf{S}_{\text{LAB}})))) \in \mathbb{R}^{T\times 3\times H\times W}. \tag{38}$$

4.3.3. Soft image reconstruction

The underwater image formation [60, 61] is governed by:

$$\mathbf{X}_{\text{img}}(x) = \mathbf{Y}(x)\,\mathbf{t}_m(x) + B(1 - \mathbf{t}_m(x)). \tag{39}$$

where $\mathbf{X}_{\text{img}}(x)$ represents observed intensity at pixel x, $\mathbf{Y}(x)$ corresponds to scene radiance or the true intensity of the object at pixel x, which we aim to recover, $\mathbf{t}_m(x)$ denotes transmission map, referring to the fraction of light that successfully reaches the optical sensor from the object at pixel x without being scattered, B is the background light or ambient light, which is the light scattered into the camera due to the medium. The pixel $x \in \{1, 2, \ldots, H \times W\}$, where H and W indicate the image's height and width. Equation (39) is rewritten to estimate scene radiance $\mathbf{Y}(x)$ by substituting $\mathbf{K}(x) = \frac{1}{\mathbf{t}_m(x)} - 1$.

$$\mathbf{Y}(x) = \mathbf{K}(x)\,\mathbf{X}_{\text{img}}(x) - \mathbf{K}(x)\,\mathbf{B} + \mathbf{X}_{\text{img}}(x). \tag{40}$$

The estimates of $\mathbf{K} \in \mathbb{R}^{T\times 3\times H\times W}$, and background light $\mathbf{B} \in \mathbb{R}^{T\times 3\times H\times W}$ are subsequently passed into soft image reconstruction block together with the hazy time-dependent input image sequence $\mathbf{X}_{\text{img}} \in \mathbb{R}^{T\times 3\times H\times W}$. The soft image reconstruction block is governed by (40) and uses the inputs to predict the dehazed image $\widehat{\mathbf{Y}}$.

4.4. Loss function

Underwater image dehazing presents unique challenges due to non-uniform light attenuation, scattering effects, and color distortions, which degrade visibility and structural integrity. Traditional loss functions such as MSE alone often fail to preserve fine details, contrast, and perceptual quality, as they primarily focus on pixel-wise intensity differences. To address these limitations, this research designs an application-specific loss function that integrates perceptual and structural quality metrics alongside conventional pixel-wise losses.

The proposed composite loss function ensures that the model effectively reduces haze while maintaining structural fidelity, preserving natural color distribution, and suppressing noise artifacts. The loss formulation combines MSE loss [62], structural similarity index measure (SSIM) loss [62], and total variance (TV) loss [63], each contributing to different aspects of image enhancement:

MSE loss: minimizes direct pixel intensity differences to ensure reconstruction consistency,

$$\mathcal{L}_{\mathrm{MSE}}\left(Y,\hat{Y}\right)=\frac{1}{H\ W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(Y_{i,j}-\hat{Y}_{i,j}\right)^{2}. \tag{41}$$

Here, H and W denotes image's height and width, Y represents the reference image, $\hat{Y}$ denotes the reconstructed image. $\hat{Y}_{i,j}$ indicates the membrane potential (intensity) at position (i,j) in the reconstructed image, and $Y_{i,j}$ represents pixel intensity at position (i,j) in the reference image.

SSIM loss: enforces structural and perceptual similarity to the reference image,

$$\mathcal{L}_{\mathrm{SSIM}}\left(Y,\hat{Y}\right)=\frac{\left(2\mu_{Y}\mu_{\hat{Y}}+C_{1}\right)\left(2\sigma_{Y\hat{Y}}+C_{2}\right)}{\left(\mu_{Y}^{2}+\mu_{\hat{Y}}^{2}+C_{1}\right)\left(\sigma_{Y}^{2}+\sigma_{\hat{Y}}^{2}+C_{2}\right)}, \tag{42}$$

where μ_{Y} and $\mu_{\hat{Y}}$ are the means, σ_{Y}^{2} and $\sigma_{\hat{Y}}^{2}$ are the variances, $\sigma_{Y\hat{Y}}$ is the covariance. C_{1} and C_{2} are constants.

TV loss: regularizes the dehazed output to maintain smoothness and suppress artifacts.

$$\mathcal{L}_{\mathrm{TV}}\left(\hat{Y}\right)=\frac{\left(\sum_{c=1}^{C}\sum_{i=1}^{H-1}\sum_{j=1}^{W-1}\left(\left(\hat{Y}_{c,i,j}-\hat{Y}_{c,i+1,j}\right)^{2}+\left(\hat{Y}_{c,i,j}-\hat{Y}_{c,i,j+1}\right)^{2}\right)\right)}{C\ \ H\ \ W}. \tag{43}$$

Net loss: the cumulative loss function is formulated as the sum of weighted components, where the weighting factors are empirically determined to balance dehazing effectiveness, perceptual similarity, and spatial smoothness,

$$\mathcal{L}_{\mathrm{net}}=\mathcal{L}_{\mathrm{MSE}}+\alpha\left(1-\mathcal{L}_{\mathrm{SSIM}}\right)+\beta\mathcal{L}_{\mathrm{TV}}, \tag{44}$$

where $\alpha=0.5$ and $\beta=0.25$ is selected with empirical analysis.

5. Results and analysis

5.1. Implementation strategy

The proposed snnTrans-DHZ architecture is implemented using SpikingJelly [64] and the PyTorch library. The model is trained on an NVIDIA A100 GPU equipped with 40GB of RAM. It undergoes 100 training epochs on the UIEB dataset and 50 epochs on the EUVP dataset. In both cases, the validation starts at the 20th epoch, allowing the model a warm-up period. To ensure optimal performance, the most effective model is selected by identifying the lowest validation loss recorded across epochs. The Adam optimizer [65] is employed with a learning rate of 0.001 to ensure stable convergence and efficient gradient updates. In all experiments, we set the smoothing parameter $\lambda=4$ and keep it fixed across datasets and training runs. This parameter controls the sharpness of the surrogate approximation around the firing threshold. All input images are scaled 512×512 for both training and inference. The timestep T is selected as 10. The snnTrans-DHZ model is trained using the surrogate gradient-based BPTT strategy (refer to section 3.2) to update the network weights, ensuring $\mathcal{L}_{\mathrm{net}}$ minimization.

The experiments in this work can be categorized into two groups. First, the benchmark datasets (UIEB and EUVP) are used for supervised learning and quantitative evaluation. The snnTrans-DHZ model is trained and evaluated using standard full-reference metrics computed against the provided reference images. Second, the real-world mission data presented in section 5.2.1 are used solely for qualitative evaluation, showcasing the generalization ability of the proposed approach in diverse underwater

Table 3. Performance evaluation of snnTrans-DHZ framework compared with UIE-SNN. The evaluation is based on algorithmic and hardware-oriented performance metrics. Both frameworks are trained and tested on UIEB and EUVP datasets. The best values are shown in bold.

		Algorithmic performance		Hardware-oriented metrics			
Dataset	Architecture	PSNR (dB) (↑)	SSIM (↑)	# Parameters (M) (↓)	SOPs (G) (↓)	Energy (J) (↓)	Energy Ratio (×) (↓)
UIEB	**snnTrans-DHZ**	**21.6773**	**0.8795**	**0.57**	**7.42**	**0.0151**	1 ×
	UIE-SNN	17.7801	0.7454	31.03	147.49	0.1327	8.79 ×
EUVP	**snnTrans-DHZ**	**23.4562**	**0.8439**	**0.57**	**8.43**	**0.0159**	1 ×
	UIE-SNN	23.1725	0.7890	31.03	84.65	0.0732	4.60 ×

scenarios. Accordingly, no training, fine-tuning, or quantitative metric computation is performed in the second scenario.

The snnTrans-DHZ model exhibits superior algorithmic performance, achieving higher PSNR and SSIM across UIEB and EUVP datasets. The hybrid RGB-LAB color space and spiking transformer architecture used in snnTrans-DHZ enhanced the feature extraction and color consistency. This is critical for underwater visual tasks where color distortions and light scattering significantly degrade image quality. The incorporation of the custom loss function $\mathcal{L}_{\text{net}}$ further optimizes perceptual quality by balancing pixel-wise prediction, structural similarity, and smoothness.

5.2. Performance evaluation

Energy consumption reported in this study is obtained via analytical estimation rather than direct measurement on physical neuromorphic hardware. Specifically, we estimate the FLOPs/SOPs, spike rate, energy, and energy ratio using analytical formulations for SNN energy modeling (refer to section 3.4). Therefore, the reported values should be interpreted as comparative analytical estimates that enable consistent efficiency comparisons across methods, rather than hardware-in-the-loop power measurements that depend on a particular neuromorphic platform, deployment configuration, and measurement setup.

A key advantage of snnTrans-DHZ is its drastically lower computational overhead. It requires only 0.57 M parameters compared to 31.03 M in UIE-SNN, making it significantly lighter and more efficient. The reduction in parameters directly impacts the computational complexity, allowing deployment on hardware with limited memory and processing capabilities, such as edge devices in underwater robotic platforms. Furthermore, snnTrans-DHZ significantly reduces the number of synaptic operations, requiring only 7.42 G and 8.43 G SOPs for UIEB and EUVP datasets, respectively, compared to 147.49 G and 84.65 G for UIE-SNN (refer to table 3).

The energy consumption of snnTrans-DHZ (0.0151 J for UIEB, and 0.0159 J for EUVP) is significantly lower than UIE-SNN, which consumes 0.1327 J and 0.0732 J respectively. The energy ratio metric further highlights the disparity, showing that snnTrans-DHZ achieves the same task with 8.79× and 4.60× less energy consumption than UIE-SNN, making it ideal for low-powered underwater robots.

The qualitative analysis of the proposed snnTrans-DHZ framework is evaluated on UIEB, the real-world underwater dataset, and is presented in figure 5. The qualitative comparison between snnTrans-DHZ and UIE-SNN highlights significant differences in image enhancement quality, particularly in terms of color restoration, structural detail preservation, and noise reduction. The visual results presented in the figure 5 demonstrate that snnTrans-DHZ produces clearer, more vibrant, and artifact-free underwater images, making it a superior choice for real-world applications.

One of the most prominent improvements in snnTrans-DHZ is its ability to restore natural colors more effectively. Underwater images frequently exhibit a noticeable green or blue tint caused by light scattering and absorption. The images processed by UIE-SNN still retain a noticeable color cast, leading to duller and less realistic appearances. In contrast, snnTrans-DHZ successfully corrects color distortions, producing images with more natural and balanced tones. This improvement is particularly beneficial for marine biology and underwater exploration applications, where accurate color representation is crucial for species identification and habitat assessment.

Beyond color restoration, structural detail preservation is another key aspect where snnTrans-DHZ excels. The zoomed-in insets in figure 5 reveal that UIE-SNN struggles to retain fine textures, often introducing blurriness and loss of detail in complex regions such as fish skin, coral structures, and shipwrecks. On the other hand, snnTrans-DHZ enhances edge sharpness and texture clarity, making objects more distinguishable. This improvement is essential for underwater archaeology and object recognition, where fine-grained details play a crucial role in analysis.

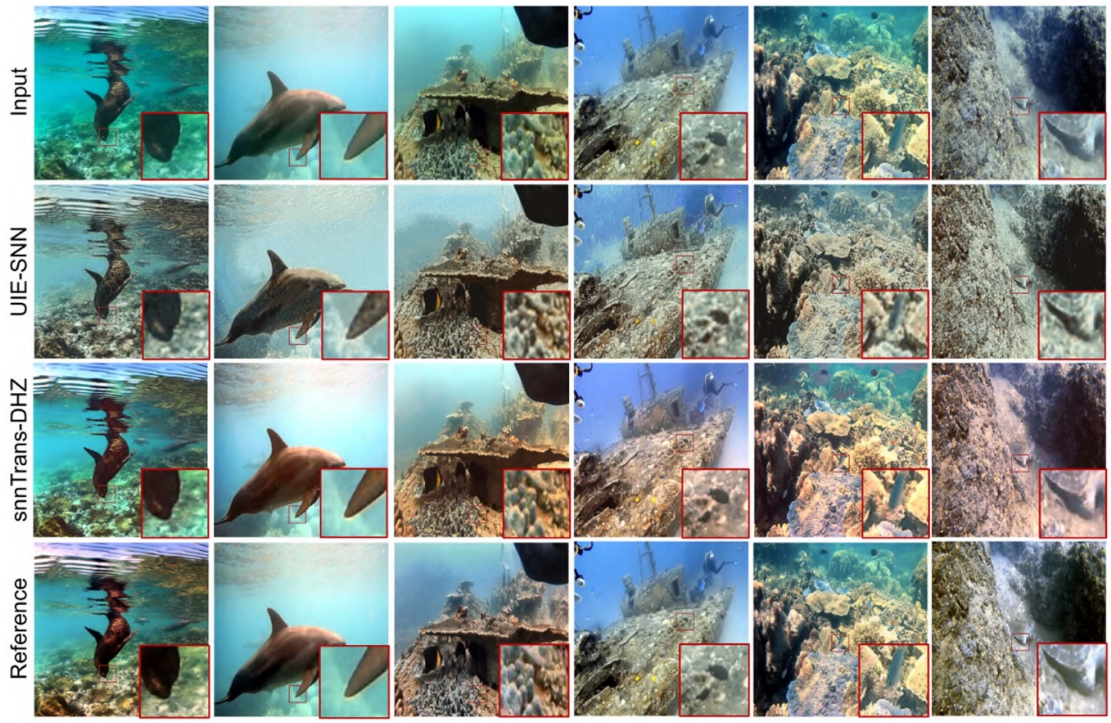

Figure 5. Qualitative analysis of the proposed snnTrans-DHZ framework on six different images from UIEB test samples. The top row indicates the unprocessed, raw underwater input images. The network-predicted images using the UIE-SNN and snnTrans-DHZ frameworks are shown in the second and third rows respectively. The last row represents the visibility-enhanced reference images. Reproduced with permission from [13].

Noise reduction and artifact suppression are additional strengths of the snnTrans-DHZ framework. Underwater images often contain artifacts due to poor visibility and environmental disturbances. The results indicate that UIE-SNN generates excessive noise, particularly in darker and smooth-surfaced regions, creating unnatural textures. In contrast, snnTrans-DHZ effectively reduces noise while maintaining sharpness, leading to more visually appealing and realistic outputs. Additionally, snnTrans-DHZ provides superior clarity in complex underwater scenes, such as shipwrecks and coral reefs. The method accurately reconstructs fine details, preserving object boundaries and intricate textures. This is particularly valuable for environmental monitoring, where capturing fine structural details is necessary for assessing coral health and detecting underwater debris.

5.2.1. Effectiveness of snnTrans-DHZ on underwater robotic missions

Figure 6 presents a qualitative analysis of snnTrans-DHZ applied to raw underwater images captured during various underwater missions, showcasing its effectiveness in enhancing underwater visual perception. The results are divided into four sections: (A) indoor pool experiments conducted at Khalifa University Marine Robotics Pool Facility, and (B)–(D) real-world underwater scenes collected during a 20-day expedition in the Arabian Gulf, in collaboration with OceanX, M42 Healthcare, and Khalifa University.

In scenario (A), images from the controlled pool environment exhibit typical underwater distortions, including greenish color casts, reduced contrast, and blurriness. The application of snnTrans-DHZ successfully enhances these images by improving color balance and contrast, making objects such as corals and robotic vehicles more visually distinguishable. This demonstrates the model's capability to function in structured testing environments, where controlled assessments of underwater imaging algorithms are crucial for real-world deployment.

The images in scenario (B) depict deep-sea intervention operations (at around 143 m depth) involving ROVs equipped with manipulators. The raw images suffer from poor lighting conditions, significant backscatter, and low visibility. The snnTrans-DHZ enhancement results show improved clarity, edge definition, and contrast, making fine details of the manipulator arms and seabed textures more visible. This enhancement is particularly beneficial for underwater inspection and intervention tasks, where operators rely on clear visual feedback to precisely manipulate objects.

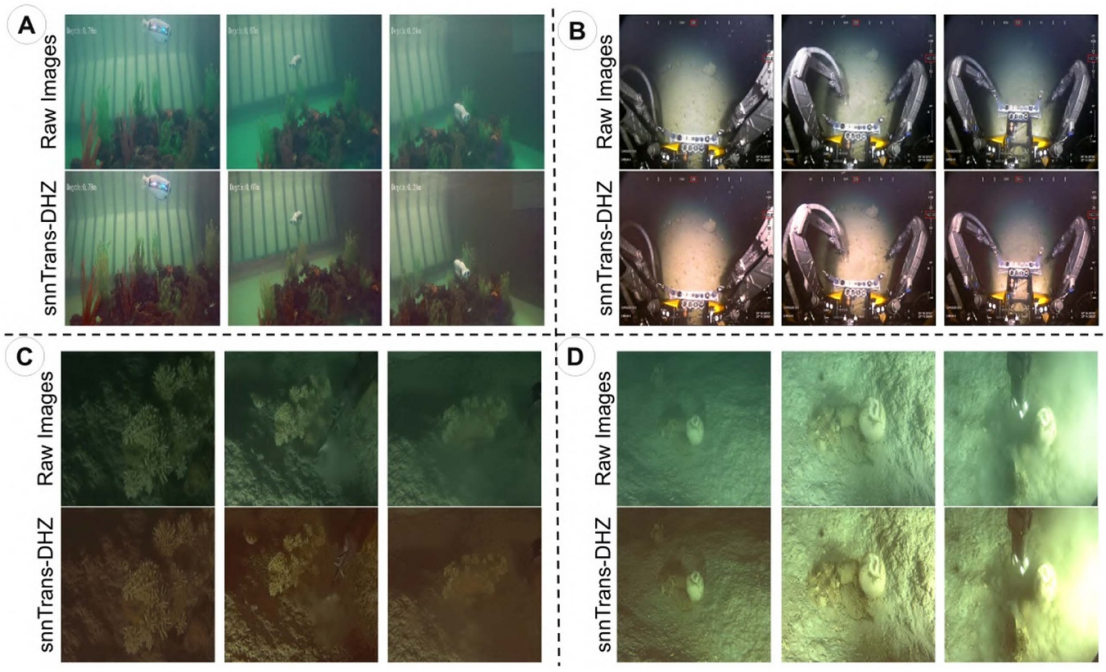

Figure 6. Qualitative analysis of snnTrans-DHZ on raw underwater images captured during various underwater missions. (A) The scenes are collected from the in-house Khalifa University Marine Robotics Pool Facility; (B)–(D) The underwater scenes collected during a 20 day expedition mission in the Arabian Gulf to explore the impacts of climate change on UAE waters (a collaboration between OceanX, M42 Healthcare, and Khalifa University).

The scenarios (C) and (D) capture marine ecosystems, particularly coral structures, and benthic organisms, affected by challenging underwater lighting conditions. The raw images in these scenes display severe color distortion and reduced visibility due to light absorption and scattering. The snnTrans-DHZ method enhances color fidelity, structural visibility, and scene interpretability, making marine biodiversity more distinguishable. Such improvements are critical for marine research and ecological monitoring, enabling more accurate assessments of coral health, habitat changes, and climate change effects on marine ecosystems.

The qualitative analyses presented in figures 5 and 6 highlight the generalization capabilities of the proposed snnTrans-DHZ framework across diverse underwater environments. While the model was trained solely on the publicly available UIEB dataset (figure 5), qualitative assessments on images from significantly different real-world underwater scenarios, including controlled pool facilities (figure 6(A)) and challenging deep-sea conditions (depths exceeding 140 m, figures 6(B)–(D), demonstrate its effectiveness. The snnTrans-DHZ maintains robust dehazing performance and perceptual clarity across varying turbidity levels and distinct lighting conditions, underscoring its practical applicability and generalization beyond the training dataset.

Overall, the qualitative results demonstrate that snnTrans-DHZ effectively enhances underwater images across diverse conditions, ranging from structured test environments to real-world deep-sea and ecological exploration missions. The proposed method significantly improves contrast, color correction, and structural clarity, making it a valuable tool for underwater robotics, marine research, and environmental monitoring.

5.3. Ablation studies

5.3.1. Threshold membrane potential

As shown in table 4, the model incorporating LTMP_{LIF} with RGB-LAB processing achieves the highest PSNR (21.6773 dB) and SSIM value 0.8795. This settings demonstrates that integrating LTMP_{LIF} neurons with hybrid color space effectively enhances model performance. These results highlight that adaptive neuron dynamics and combined luminance-chromaticity processing significantly contribute to better haze removal and perceptual image quality, without substantially increasing model complexity.

Replacing adaptive LTMP_{LIF} neurons with fixed-threshold neurons while retaining RGB-LAB hybrid processing leads to a noticeable performance drop, with PSNR and SSIM decreasing to 20.8279 dB and, 0.8722 respectively. This reduction indicates the pivotal role played by the adaptive threshold in LTMP_{LIF} neurons, emphasizing their contribution to modeling temporal dynamics and improving the overall

Table 4. Evaluation of snnTrans-DHZ under different ablation settings. The best values are shown in bold.

Ablation settings				PSNR (dB) (↑)	SSIM (↑)	# Parameters (M)(↓)
Threshold		Color space				
Fixed (0.5 V)	Adaptive	RGB-LAB	RGB-Alone			
	✓	✓		**21.6773**	**0.8795**	0.57
✓		✓		20.8279	0.8722	0.57
	✓		✓	18.2622	0.6764	**0.29**

Table 5. Performance evaluation of the snnTrans-DHZ framework on different image resolutions.

Image resolution	PSNR (dB) (↑)	SSIM (↑)	SOPs (G) (↓)	Energy (J) (↓)
256 × 256	21.6988	0.8818	1.6711	0.0017
512 × 512	21.6773	0.8795	7.4156	0.0150
1024 × 1024	21.6587	0.8804	25.8353	0.0266

quality of underwater image reconstruction. The consistent number of parameters confirms the performance variation arises solely from adaptive neuron mechanisms rather than model complexity.

5.3.2. Effect of hybrid color space processing

It is evident from table 4 that processing images with a hybrid RGB-LAB color space substantially improves the dehazing performance of the model. The results clearly demonstrate the critical advantage provided by hybrid RGB-LAB processing, suggesting that separately handling luminance and chromaticity channels greatly enhances dehazing and image visibility. The reduced parameter count (0.29 M) here is due to the omission of the LAB processing branch, underscoring that the observed performance drop originates primarily from the absence of hybrid color space processing rather than the adaptive LTMP_{LIF} mechanism.

5.3.3. Image resolution

To evaluate the efficiency and scalability of snnTrans-DHZ for underwater robotics applications, we analyze its performance across different input image resolutions. The results, summarized in table 5, provide insights into the trade-offs between image quality, computational complexity, and energy consumption at varying resolutions.

The PSNR and SSIM values remain relatively stable across resolutions, indicating that snnTrans-DHZ maintains high-quality image enhancement regardless of input size. The PSNR fluctuates slightly around 21.68 dB, and the SSIM consistently exceeds 0.88, demonstrating the model's robustness in restoring underwater images with minimal degradation, even at lower resolutions. This is crucial for underwater robotics, where efficient processing of lower-resolution images can significantly reduce latency without compromising perceptual quality.

However, increasing resolution leads to a sharp rise in computational cost. The SOPs grow from 1.6711 G at 256 × 256 image resolution to 25.8353 G at 1024 × 1024 image resolution, demonstrating a nearly 15.46 × increased number of synaptic operations. Similarly, energy consumption rises from 0.0017 J to 0.0266 J, highlighting the increased computational burden of higher-resolution inputs. While lower-resolution image processing is often preferred for real-time underwater robotics applications due to its efficiency, there are critical scenarios where high-resolution image processing becomes essential. Tasks such as marine biodiversity monitoring, archaeological site mapping, deep-sea exploration, and defect detection in underwater infrastructure require fine-grained visual details to ensure accurate interpretation and analysis. In such cases, the ability to process high-resolution images efficiently is a key advantage.

For practical deployment in underwater robots, 512 × 512 resolution emerges as the optimal choice, offering a strong balance between quality (PSNR = 21.6773 dB, SSIM = 0.8795) and efficiency (SOPs = 7.4156 G, energy = 0.0150 J). This resolution provides sufficient details for object detection and navigation while maintaining a manageable computational load, facilitating real-time processing in energy-constrained environments.

These findings highlight the scalability of snnTrans-DHZ, making it adaptable for a range of underwater robotics applications. Depending on mission requirements, lower resolutions can be leveraged for

Table 6. Performance evaluation of the proposed snnTrans-DHZ framework with SOTA UIE methods trained and tested using the UIEB dataset. The best values are shown in bold.

Method	PSNR (dB)(↑)	SSIM (↑)	#Parameters (↓)	Number of operations[a] (↓)	Energy (J) (↓)	Energy ratio [b] (×) (↓)
CNN-based method:						
WaterNet [13]	21.1700	0.8689	1.09	285.80	1.3147	87.07×
Shallow-UWNet [2]	18.2780	0.8550	0.22	216.14	0.9942	65.84×
Ucolor [20]	18.8400	0.7612	157.42	887.80	4.0839	270.46×
PUIE-Net [14]	21.3800	0.882	1.40	300.04	1.3802	91.40×
PDCFNet [15]	**27.3700**	0.9200	1.82	172.77	0.7947	52.63×
UIEConv [16]	24.1197	0.9283	3.31	243.06	1.1181	74.05×
GAN-based method:						
PUGAN [17]	19.9700	0.7848	95.66	144.18	0.6632	43.92×
LFT-DGAN [19]	24.4591	0.8284	3.30	264.28	1.2157	80.51×
Attention-based method:						
LANet [21]	24.0600	0.9085	5.15	355.37	1.6347	108.26×
Deep-WaveNet [22]	23.3000	0.9199	—	145.18	0.6678	44.22×
DICAM [23]	24.4300	**0.9375**	—	106.26	0.4888	32.37×
Transformer-based method:						
U-Trans [4]	22.9100	0.9100	65.60	132.04	0.6074	40.22×
MFMN [25]	24.2067	0.8992	**0.20**	205.61	0.9458	62.64×
MMIETransformer [10]	23.2800	0.9100	16.80	164.00	0.7544	49.96×
WPFNet [28]	22.3400	0.9072	2.16	40.20	0.1849	12.24×
DDformer [11]	23.5007	0.8944	7.58	35.74	0.1644	10.89×
Ghost-UNet [9]	23.1600	0.8300	1.68	10.72	0.0493	3.26×
Dynamic SpectraFormer [29]	25.9800	0.9341	64.20	100.40	0.4618	30.58×
CFPNet [27]	25.6900	0.8860	47.84	111.84	0.5145	34.07×
SNN-based method:						
UIE-SNN [12]	17.7801	0.7454	31.03	147.49	0.1327	8.79×
snnTrans-DHZ [Ours]	21.6773	0.8795	0.57	**7.4156**	**0.0151**	**1×**

[a] The number of operations is measured in GSOPs for SNN-based methods and in GFLOPs for others.

[b] The energy ratio is calculated as $\frac{E_{\text{method}}}{E_{\text{snnTrans}-\text{DHZ}}}$

fast, low-power processing, while higher resolutions may be used selectively when detailed scene understanding is necessary.

6. Discussions

The optimization objective in snnTrans-DHZ targets a balanced trade-off between computational energy efficiency and image enhancement quality. While the composite loss is designed to preserve perceptual structure and suppress artifacts, the overall model design additionally prioritizes energy efficiency through spike-based sparse computation. Consequently, the goal of snnTrans-DHZ is not to maximize absolute image quality at any computational cost. Instead, it aims to achieve reliable reconstruction quality under constrained model complexity and energy usage, which is essential for resource-limited deployment scenarios.

The comparison between the proposed snnTrans-DHZ algorithm and SOTA learning-based UIE methods is presented in table 6. The comparison of snnTrans-DHZ with SOTA UIE methods highlights its balance between performance and efficiency. While snnTrans-DHZ yields PSNR score of 21.6773 dB and SSIM as 0.8795, which is lower than leading transformer-based methods like Dynamic SpectraFormer (PSNR of 25.98 dB, SSIM of 0.9341) and CFPNet (PSNR of 25.69 dB, SSIM of 0.8860). The proposed snnTrans-DHZ framework significantly outperforms other compact models in computational efficiency. The higher image quality from transformer-based models comes at the cost of significantly higher computational complexity and energy consumption, making them less suitable for resource-constrained environments.

A key advantage of snnTrans-DHZ is its lightweight design, with only 0.57 M parameters, making it one of the most compact models in the comparison. In contrast, Dynamic SpectraFormer and CFPNet require 64.2 M and 47.84 M parameters, respectively, making them 112× and 84× larger. Even the lightweight MFMN model, which has a smaller parameter count (0.20 M), still consumes significantly higher

energy. This compact architecture of SNNTrans-DHZ enables efficient deployment on edge devices without the memory constraints of larger transformer-based networks.

Another critical metric is computational cost. snnTrans-DHZ requires only 7.4156 G SOPs, whereas Dynamic SpectraFormer (100.4 G FLOPs) and CFPNet (111.84 G FLOPs) are 13.54 × and 15.08 × more computationally intensive, respectively. Even relatively efficient transformer-based methods like DDformer (35.74 G FLOPs) and Ghost-UNet (10.72 G FLOPs) remain at least 4.82 × and 1.44 × more computationally demanding than snnTrans-DHZ. This drastic reduction in computational requirements allows snnTrans-DHZ to operate with notably lower latency, making it well-optimized for real-time underwater robotic applications.

Energy efficiency is another defining strength of SNNTrans-DHZ. It consumes only 0.0151 J, which is substantially lower than all other SOTA UIE methods. Ghost-UNet, the most energy-efficient transformer-based model, requires 0.0493 J, which is 3.26 × more than snnTrans-DHZ. Dynamic SpectraFormer (0.4618 J) and CFPNet (0.5145 J) require 30.58 × and 34.07 × more energy, respectively. This significant energy savings extends the operational lifespan of battery-powered underwater robotics, making snnTrans-DHZ an optimal choice for long-duration missions.

Some CNN-based models, such as PDCFNet (27.37 dB, 0.9200 SSIM) and UIEConv (24.12 dB, 0.9283 SSIM), achieve higher PSNR and SSIM values. However, they still exhibit considerably higher energy consumption and computational costs. For example, PDCFNet consumes 0.7947 J (52.63 × more than snnTrans-DHZ, while UIEConv requires 1.1181 J (74.05 × more). Thus, snnTrans-DHZ offers a compelling alternative that balances image quality, efficiency, and deployability.

In summary, while snnTrans-DHZ does not achieve the absolute highest image enhancement quality, it significantly outperforms all competing models in both computational efficiency and energy savings. Its lightweight architecture, low power consumption, and ability to perform real-time processing make it highly practical for underwater robotic applications, where resource constraints are a critical factor. This substantial energy savings makes snnTrans-DHZ a more practical solution for battery-powered underwater robotics, extending mission durations while maintaining reliable enhancement quality.

7. Conclusions and future research directions

This article detailed the development of snnTrans-DHZ, a lightweight and energy-efficient spiking transformer-based framework for underwater image dehazing. The snnTrans-DHZ algorithm achieves PSNR of 21.6773 dB and SSIM of 0.8795 on UIEB, and PSNR of 23.4562 dB and SSIM of 0.8439 on EUVP. The total network parameters in the snnTrans-DHZ architecture are **0.5670 M**. Compared to the existing SOTA image enhancement methods, the snnTrans-DHZ architecture could significantly reduce the model complexity and energy consumption. Future research will focus on deploying the snnTrans-DHZ algorithm onto an edge device and analyzing its applicability in real-world deployments. Since the number of parameters is less than one million, it can be deployed on the Loihi neuromorphic chip to evaluate its real-time performance on neuromorphic hardware.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

This work is supported by Khalifa University under Award No. RC1-2018-KUCARS-8474000136. We express our gratitude to Dr Fakhreddine Zayer for his valuable feedback and insightful suggestions during the preparation of this manuscript.

Author contributions

Vidya Sudevan 0009-0002-2734-624X
Conceptualization (lead), Data curation (lead), Formal analysis (lead), Methodology (lead), Software (lead), Validation (lead), Visualization (lead), Writing – original draft (lead)

Rizwana Kausar 0009-0009-9694-0737
Formal analysis (supporting), Visualization (supporting), Writing – review & editing (supporting)

Sajid Javed 0000-0002-0036-2875
Formal analysis (supporting), Investigation (supporting), Resources (supporting), Supervision (supporting), Writing – review & editing (supporting)

Hamad Karki 0000-0003-3577-915X
Formal analysis (supporting), Investigation (supporting), Project administration (supporting), Resources (supporting), Supervision (supporting), Writing – review & editing (supporting)

Giulia De Masi 0000-0003-3284-880X
Formal analysis (supporting), Investigation (supporting), Project administration (supporting), Resources (supporting), Supervision (supporting), Writing – review & editing (supporting)

Jorge Dias 0000-0002-2725-8867
Funding acquisition (lead), Project administration (lead), Resources (lead), Supervision (lead), Writing – review & editing (lead)

References

[1] Zhou J, Yang T and Zhang W 2023 Underwater vision enhancement technologies: a comprehensive review, challenges and recent trends *Appl. Intell.* **53** 3594–621
[2] Naik A, Swarnakar A and Mittal K 2021 Shallow-uwnet: compressed model for underwater image enhancement (student abstract) *Proc. AAAI Conf. on Artificial Intelligence* vol 35 pp 15853–4
[3] Jie Li, Skinner K A, Eustice R M and Johnson-Roberson M 2017 Watergan: unsupervised generative network to enable real-time color correction of monocular underwater images *IEEE Robot. Autom. Lett.* **3** 387–94
[4] Peng L, Zhu C and Bian L 2023 U-shape transformer for underwater image enhancement *IEEE Trans. Image Process.* **32** 3066–79
[5] Chengqin W, Shuai Y, Qingson H, Jingxiang X and Zhang L 2024 Underwater image enhancement via dehazing and color restoration (arXiv:2409.09779)
[6] Roy K, Jaiswal A and Panda P 2019 Towards spike-based machine intelligence with neuromorphic computing *Nature* **575** 607–17
[7] Hebei Li, Zhang Y, Xiong Z and Sun X 2024 Deep multi-threshold spiking-UNet for image processing *Neurocomputing* **586** 127653
[8] Castagnetti A, Pegatoquet A and Miramond B 2023 Spiden: deep spiking neural networks for efficient image denoising *Front. Neurosci.* **17** 1224457
[9] Sun L, Wenqing Li and Yingjie X 2024 Ghost-unet: lightweight model for underwater image enhancement *Eng. Appl. Artif. Intell.* **133** 108585
[10] Kulkarni A, Phutke S S, Vipparthi S K and Murala S 2024 Multi-medium image enhancement with attentive deformable transformers *IEEE Trans. Emerg. Topics Comput. Intell.* **9** 1659–72
[11] Gao Z, Yang J, Jiang F, Jiao X, Dashtipour K, Gogate M and Hussain A 2024 Ddformer: dimension decomposition transformer with semi-supervised learning for underwater image enhancement *Knowl.-Based Syst.* **297** 111977
[12] Sudevan V, Zayer F, Kausar R, Javed S, Karki H, De Masi G and Dias J 2025 Underwater image enhancement by convolutional spiking neural networks (arXiv:2503.20485)
[13] Chongyi Li, Guo C, Ren W, Cong R, Hou J, Kwong S and Tao D 2020 An underwater image enhancement benchmark dataset and beyond *IEEE Trans. Image Process.* **29** 4376–89
[14] Zhenqi F, Wang W, Huang Y, Ding X and Kai-Kuang M 2022 Uncertainty inspired underwater image enhancement *European Conf. on Computer Vision* (Springer) pp 465–82
[15] Zhang S, Daoliang Li, and Zhao R 2024 PDCFnet: enhancing underwater images through pixel difference convolution and cross-level feature fusion (arXiv:2409.19269)
[16] Dazhao D, Li E, Lingyu S, Fanjiang X, Niu J and Sun F 2024 A physical model-guided framework for underwater image enhancement and depth estimation (arXiv:2407.04230)
[17] Cong R, Yang W, Zhang W, Chongyi Li, Guo C-L, Huang Q and Kwong S 2023 Pugan: physical model-guided underwater image enhancement using GAN with dual-discriminators *IEEE Trans. Image Process.* **32** 4472–85
[18] Gonzalez-Sabbagh S, Robles-Kelly A and Gao S 2024 DGD-cGAN: a dual generator for image dewatering and restoration *Pattern Recognit.* **148** 110159
[19] Zheng S, Wang R, Zheng S, Wang L and Liu Z 2024 A learnable full-frequency transformer dual generative adversarial network for underwater image enhancement *Front. Mar. Sci.* **11** 1321549
[20] Chongyi Li, Anwar S, Hou J, Cong R, Guo C and Ren W 2021 Underwater image enhancement via medium transmission-guided multi-color space embedding *IEEE Trans. Image Process.* **30** 4985–5000
[21] Liu S, Fan H, Lin S, Wang Q, Ding N and Tang Y 2022 Adaptive learning attention network for underwater image enhancement *IEEE Robot. Autom. Lett.* **7** 5326–33
[22] Sharma P, Bisht I and Sur A 2023 Wavelength-based attributed deep neural network for underwater image restoration *ACM Trans. Multimedia Comput. Commun. Appl.* **19** 1–23
[23] Tolie H F, Ren J and Elyan E 2024 Dicam: deep inception and channel-wise attention modules for underwater image enhancement *Neurocomputing* **584** 127585
[24] Wang B, Xu H, Jiang G, Yu M, Ren T, Luo T and Zhu Z 2024 Uie-convformer: underwater image enhancement based on convolution and feature fusion transformer *IEEE Trans. Emerg. Top. Comput. Intell.* textbf8 1952–68
[25] Zheng S, Wang R, Zheng S, Wang F, Wang L and Liu Z 2024 A multi-scale feature modulation network for efficient underwater image enhancement *J. King Saud Univ. Comput. Inf. Sci.* **36** 101888
[26] Nie X, Pan S, Zhai X, Tao S, Fengzhong Q, Wang B, Huilin G and Xiao G 2024 Image-conditional diffusion transformer for underwater image enhancement (arXiv:2407.05389)
[27] Xianping F, Qin W, Li F, Xu F and Yan X 2024 Cfpnet: complementary feature perception network for underwater image enhancement *IEEE J. Ocean. Eng.* **50** 150–63

[28] Liu S, Fan H, Wang Q, Han Z, Guan Y and Tang Y 2024 Wavelet–pixel domain progressive fusion network for underwater image enhancement *Knowl.-Based Syst.* **299** 112049
[29] Zhiqiang H, Tao Y, Huang S and Ishikawa M 2024 Dynamic spectraformer for ultra-high-definition underwater image enhancement *2024 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (IEEE) pp 8634–41
[30] Li H, Zhang Y, Xiong Z, Zha Z-J, and Sun X 2023 Deep spiking-UNet for image processing (arXiv:2307.10974)
[31] Diehl P U, Neil D, Binas J, Cook M, Liu S-C and Pfeiffer M 2015 Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing *2015 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 1–8
[32] Yan Z, Zhou J and Wong W-F 2021 Near lossless transfer learning for spiking neural networks *Proc. AAAI Conf. on Artificial Intelligence* vol 35 pp 10577–84
[33] Diehl P U and Cook M 2015 Unsupervised learning of digit recognition using spike-timing-dependent plasticity *Front. Comput. Neurosci.* **9** 99
[34] Liu Q, Pan G, Ruan H, Xing D, Xu Q and Tang H 2020 Unsupervised aer object recognition based on multiscale spatio-temporal features and spiking neurons *IEEE Trans. Neural Netw. Learn. Syst.* **31** 5300–11
[35] Liu D, Bellotto N and Yue S 2019 Deep spiking neural network for video-based disguise face recognition based on dynamic facial movements *IEEE Trans. Neural Netw. Learn. Syst.* **31** 1843–55
[36] Lee J H, Delbruck T and Pfeiffer M 2016 Training deep spiking neural networks using backpropagation *Front. Neurosci.* **10** 508
[37] Yujie W, Deng L, Li G, Zhu J and Shi L 2018 Spatio-temporal backpropagation for training high-performance spiking neural networks *Front. Neurosci.* **12** 331
[38] Chakraborty B, She X and Mukhopadhyay S 2021 A fully spiking hybrid neural network for energy-efficient object detection *IEEE Trans. Image Process.* **30** 9014–29
[39] Roy D, Panda P and Roy K 2019 Synthesizing images from spatio-temporal representations using spike-based backpropagation *Front. Neurosci.* **13** 621
[40] Comșa I-M, Versari L, Fischbacher T and Alakuijala J 2021 Spiking autoencoders with temporal coding *Front. Neurosci.* **15** 712667
[41] Cao J, Wang Z, Guo H, Cheng H, Zhang Q and Renjing X 2024 Spiking denoising diffusion probabilistic models *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision* pp 4912–21
[42] Song T, Jin G, Li P, Jiang K, Chen X, and Jin J 2024 Learning a spiking neural network for efficient image deraining (arXiv:2405.06277)
[43] Chen X, Yang Q, Jibin W, Haizhou Li and Tan K C 2023 A hybrid neural coding approach for pattern recognition with spiking neural networks *IEEE Trans. Pattern Anal. Mach. Intell.* **46** 3064–78
[44] Dupeyroux J, Stroobants S and De Croon G C H E 2022 A toolbox for neuromorphic perception in robotics *2022 8th Int. Conf. on Event-Based Control, Communication and Signal Processing (EBCCSP)* (IEEE) pp 1–7
[45] Kiselev M 2016 Rate coding vs. temporal coding-is optimum between? *2016 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 1355–9
[46] Hwang S and Kung J 2024 One-spike snn: single-spike phase coding with base manipulation for ann-to-snn conversion loss minimization *IEEE Trans. Emerg. Top. Comput.* **13** 162–72
[47] Kim J, Kim H, Huh S, Lee J and Choi K 2018 Deep neural networks with weighted spikes *Neurocomputing* **311** 373–86
[48] Kim Y, Park H, Moitra A, Bhattacharjee A, Venkatesha Y and Panda P 2022 Rate coding or direct coding: Which one is better for accurate, robust and energy-efficient spiking neural networks? *ICASSP 2022-2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 71–75
[49] Auge D, Hille J, Mueller E and Knoll A 2021 A survey of encoding techniques for signal processing in spiking neural networks *Neural Process. Lett.* **53** 4693–710
[50] Mukhoty B, Bojkovic V, de Vazelhes W, Zhao X, De Masi G, Xiong H and Bin G 2023 Direct training of snn using local zeroth order method *Advances in Neural Information Processing Systems* vol 36 pp 18994–9014
[51] Eshraghian J K, Ward M, Neftci E O, Wang X, Lenz G, Dwivedi G, Bennamoun M, Jeong D S and Wei D L 2023 Training spiking neural networks using lessons from deep learning *Proc. IEEE* **111** 1016–54
[52] Javanshir A, Thanh Thi Nguyen M A P M and Kouzani A Z 2022 Advancements in algorithms and neuromorphic hardware for spiking neural networks *Neural Comput.* **34** 1289–328
[53] Yan Z, Zhou J and Wong W-F 2023 Cq^{+}+ training: minimizing accuracy loss in conversion from convolutional neural networks to spiking neural networks *IEEE Trans. Pattern Anal. Mach. Intell.* **45** 11600–11
[54] Yangfan H, Zheng Q, Jiang X and Pan G 2023 Fast-snn: fast spiking neural network by converting quantized ann *IEEE Trans. Pattern Anal. Mach. Intell.* **45** 14546–62
[55] Yang Z, Guo S, Fang Y, Zhaofei Y and Liu J K 2024 Spiking variational policy gradient for brain inspired reinforcement learning *IEEE Trans. Pattern Anal. Mach. Intell.* **47** 1975–90
[56] Horowitz M 2014 1.1 computing's energy problem (and what we can do about it) *2014 IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)* (IEEE) pp 10–14
[57] Kim Y, Chough J and Panda P 2022 Beyond classification: directly training spiking neural networks for semantic segmentation *Neuromorph. Comput. Eng.* **2** 044015
[58] Leon K, Mery D, Pedreschi F and Leon J 2006 Color measurement in l* a* b* units from rgb digital images *Food Res. Int.* **39** 1084–91
[59] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Łukasz and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* p 30
[60] Jaffe J S 1990 Computer modeling and the design of optimal underwater imaging systems *IEEE J. Ocean. Eng.* **15** 101–11
[61] Akkaynak D and Treibitz T 2018 A revised underwater image formation model *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 6723–32
[62] Zhao H, Gallo O, Frosio I and Kautz J 2016 Loss functions for image restoration with neural networks *IEEE Trans. Comput. Imaging* **3** 47–57
[63] Rudin L I, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Physica. D* **60** 259–68
[64] Fang W, Chen Y, Ding J, Zhaofei Y, Masquelier T, Chen D, Huang L, Zhou H, Li G and Tian Y 2023 Spikingjelly: an open-source machine learning infrastructure platform for spike-based intelligence *Sci. Adv.* **9** eadi1480
[65] Kingma D P and Jimmy B 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)