Neuronal Self-Adaptation Enhances Capacity and Robustness of Representation in Spiking Neural Networks

Zhuobin Yang[1]* **Liao Yu**[2]* **Yeyao Bao**[1] **Jian Zhang**[1] **Liangfu Lu**[1]†
Daqing Guo[3]† **Xiaohong Li**[1]† **Yunliang Zang**[1]†
[1]Academy of Medical Engineering and Translational Medicine, Tianjin University
[2]School of Mathematical Sciences, Beihang University
[3]School of Life Science and Technology, University of Electronic Science and Technology of China
{zhuobinyang,yeyaobao_,jian_zhang,liangfulv,xhli18,yunliangzang}@tju.edu.cn
yuliao_16@buaa.edu.cn
dqguo@uestc.edu.cn

Abstract

Spiking Neural Networks (SNNs) are promising for energy-efficient, real-time edge computing, yet their performance is often constrained by the limited adaptability of conventional leaky integrate-and-fire (LIF) neurons. Existing LIF models struggle with restricted information capacity and susceptibility to noise, leading to degraded accuracy and compromised robustness. Inspired by the dynamic self-regulation of biological potassium channels, we propose the Potassium-regulated LIF (KvLIF) neuron model. KvLIF introduces an auxiliary conductance state that integrates membrane potential and spiking history to adaptively modulate neuronal excitability and reset dynamics. This design extends the dynamic response range of neurons to varying input intensities and effectively suppresses noise-induced spikes. We extensively evaluate KvLIF on both static image and neuromorphic datasets, demonstrating consistent improvements in classification accuracy and superior robustness compared to existing LIF models. Our work bridges biological plausibility with computational efficiency, offering a neuron model that enhances SNN performance while maintaining suitability for low-power neuromorphic deployment.

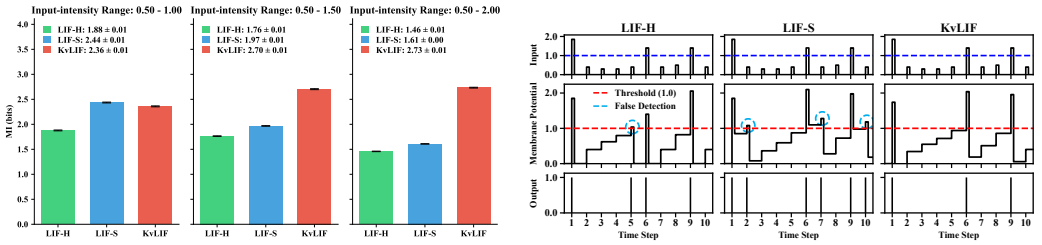
1 Introduction

Spiking Neural Networks (SNNs) offer biological plausibility, event-driven computation, and strong spatiotemporal processing, with significant energy efficiency on neuromorphic hardware for low-power, real-time applications [1–3]. The Leaky Integrate-and-Fire (LIF) neuron model is a core component for its simplicity and biological relevance [4]. However, its non-differentiable spiking hinders direct backpropagation, motivating surrogate gradient (SG) methods [5, 6]. Membrane potential reset plays a key role in performance: soft reset preserves residual temporal information, while hard reset discards it [5, 7]. Although soft reset is generally advantageous in preserving temporal information, conventional LIF models still face limitations that constrain their effectiveness in complex temporal tasks.

In SNNs, input signals are typically encoded as spike trains whose firing rates increase with input intensity [8]. Input intensities in practical settings often span a broad range (see examples in Figure

*Equal contribution
†Corresponding authors

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Input-output MI across input-intensity ranges.

(b) Neuronal responses to temporal input.

Figure 1: KvLIF enhances dynamic response range and suppresses redundant spikes. LIF-S: LIF model with soft reset; LIF-H: LIF model with hard reset.

5, Appendix A). Neuron models are therefore expected to produce responses that remain discriminable across a wide dynamic range. However, as quantified by input-output mutual information (MI), conventional LIF neurons exhibit limited information capacity at high input intensities (Figure 1(a)). When the upper bound of the input range is increased to 2.00, MI decreases for both hard- and soft-reset variants, consistent with response saturation and diminished information transmission in the high-intensity regime. Conventional LIF neurons also tend to generate redundant spikes under high-intensity inputs because membrane integration continues after salient events (Figure 1(b)). After an event-triggered spike, residual or noise-driven inputs can still accumulate and spuriously elicit additional spikes. This effect is particularly pronounced under soft reset. Together, these observations suggest that conventional LIF dynamics have restricted information capacity and limited excitability control under noisy, high-intensity temporal inputs.

Although LIF models resemble neuronal spiking, biological neurons exhibit more dynamic spiking responses due to the influence of diverse biological factors. In particular, neurons contain diverse ion currents that enable self-adaptive behavior under varying input signal intensities [9, 10]. Even within the same type of ion channel, different components can exhibit distinct properties. As shown in Figure 2(a), A-type potassium currents activate within the subthreshold potential range, dampening neuronal excitability to prevent premature action potential generation [11]. In contrast, components like delayed-rectifier potassium currents activate only at higher membrane potentials, primarily driving membrane repolarization and contributing to the subsequent afterhyperpolarization [12]. These ion channels can modulate their activation and deactivation states based on real-time membrane potential and spiking experience, thereby dynamically adjusting a neuron's response range and contributing to working memory [13].

In this work, inspired by the adaptive neuronal responses, we propose a biologically inspired Potassium Regulated Leaky Integrate-and-Fire (KvLIF) model. By dynamically adjusting its sensitivity range based on spiking history and real-time membrane potential, KvLIF achieves robust spike encoding across a wide range of input intensities in static signals and reduces spurious event detections in temporally rich data (Figure 1). The KvLIF model introduces an auxiliary state variable, $K[t]$. This variable integrates historical membrane potential fluctuations and spiking activity to adaptively modulate neuronal excitability. During the subthreshold phase, $K[t]$, inspired by A-type potassium channels, imposes voltage-dependent input shunting to suppress membrane depolarization. Upon spike generation, $K[t]$ drives a dynamic voltage reset mechanism, drawing on the functional role of delayed rectifier potassium channels in biological afterhyperpolarization. Together, $K[t]$ incorporates both membrane potential and spiking experience into the model to adjust neuronal response. This design provides several computational benefits: extended dynamic range, noise robustness, and shorter inference windows, when compared to the conventional LIF models.

The main contributions of this work are as follows:

- **Analysis of LIF model limitations**: We provide a detailed investigation of the fundamental limitations of conventional LIF neuron models, including restricted response ranges and vulnerability to noise.
- **A biologically inspired adaptive neuron**: Motivated by self-regulatory mechanisms in biological neurons, we propose the KvLIF model which features a novel two-stage dynamic regulation mechanism. It first adaptively modulates neuronal states according to membrane potential fluctuations, and then reshapes neuronal excitability based on spiking history. By

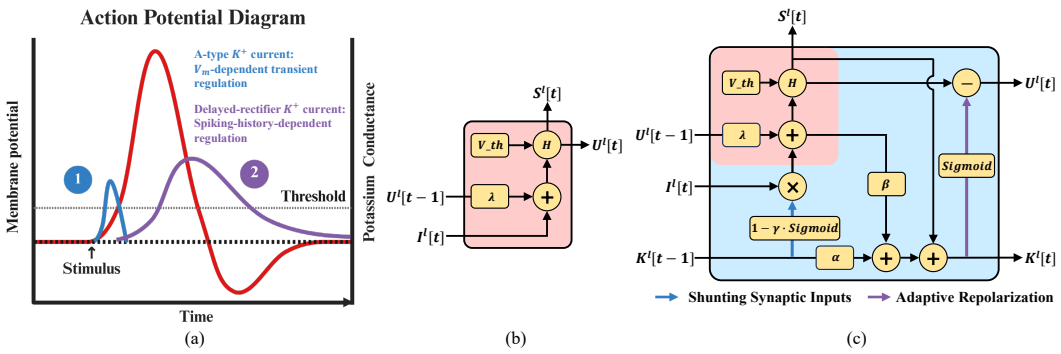


Figure 2: Biophysical motivation and computational realization of potassium-regulated neuronal self-adaptation. (a) A-type and delayed-rectifier potassium currents regulate subthreshold excitability and post-spike repolarization in biological neurons. (b) Conventional LIF performs fixed integration and reset without activity-dependent regulation. (c) KvLIF introduces a potassium conductance state $K[t]$ to coordinate synaptic shunting and adaptive repolarization, enabling voltage- and spike-history-dependent self-regulation.

coupling voltage-driven immediate regulation with spike-history-dependent post-spike regulation, KvLIF achieves a more flexible response profile and improved robustness.

- **Comprehensive empirical validation**: We perform extensive experiments across both static and neuromorphic datasets, demonstrating that the proposed KvLIF model consistently surpasses existing neuron models in classification accuracy and robustness to noise.

2 Related Work

2.1 Parametric Optimizations of Spiking Neurons

Conventional LIF neurons are computationally efficient but limited in modeling complex spatiotemporal dynamics. Optimizing neuronal internal parameters has therefore become an effective way to process multi-scale temporal information. For instance, PLIF introduces learnable membrane time constants to adapt integration dynamics through gradient-based training [14], while LTMD employs learnable thresholds to regulate neuronal firing sensitivity [15]. Beyond single-parameter optimization, expanding the structured parameter space further improves neuronal expressiveness. GLIF incorporates learnable gating factors to integrate diverse biological features and control information flow [16], whereas TC-LIF adopts a two-compartment architecture with coupling parameters to better capture long-term temporal dependencies [17].

Despite enhanced representational capacity, these parametric methods produce neuronal dynamics that remain static during inference [18], limiting adaptability and robustness in complex dynamic environments.

2.2 Adaptation Mechanisms in Spiking Neurons

Various dynamic adaptation mechanisms have proven effective in enhancing network representational capability and gradient propagation efficiency. Spike-frequency adaptation (SFA) LIF neurons, such as ALIF [19] and adLIF [20], introduce an additional adaptation variable to encode temporal context and enhance temporal processing. To alleviate gradient vanishing, Huang et al. proposed the CLIF neuron model [21], which adds a complementary membrane potential to provide extra backpropagation paths and preserve temporal gradients. Similarly, ILIF introduces two independent inhibitory variables to adaptively balance gradient preservation and neuronal overactivation [22]. From the perspective of information retention, Huang et al. designed AR-LIF, which combines adaptive reset with dynamic thresholds to reduce information loss [23]. To capture more complex spatiotemporal dependencies, Wang and Yu proposed STC-LIF, where learnable autaptic circuits regulate current inputs and historical states to improve predictive capability [18].

Current adaptive neuron models mainly rely on independent, single-stage adaptation [19–21]. For example, ALIF implements spike-frequency adaptation with a spike-history-dependent threshold that rises after spikes and suppresses firing. KvLIF instead introduces a potassium-conductance-like state $K[t]$, driven by both membrane-potential fluctuations and spike history, that shapes both membrane integration and reset. By coupling subthreshold voltage dependence with post-spike repolarization, KvLIF enables more flexible and robust neuronal dynamics, thereby achieving superior performance over existing adaptive neuron models.

3 Preliminary

3.1 LIF Neuron Model

The LIF model serves as the foundational computational unit in SNNs. As shown in Figure 2(b), the LIF neuron maintains an internal membrane potential that evolves over time. The subthreshold dynamics for neurons in layer l at time step t are governed by the accumulation of postsynaptic current $I^l[t]$ and the decay of their previous membrane potential $U^l[t-1]$. The membrane potential update is expressed as:

$$I^l[t] = W^l S^{l-1}[t] \tag{1}$$

$$U^l[t] = \lambda U^l[t-1] + I^l[t] \tag{2}$$

Here, $\lambda \in (0,1)$ is the membrane decay constant, and W^l denotes the synaptic weights from layer $l-1$ to layer l. $S^{l-1}[t]$ represents the spikes of layer $l-1$ neurons at time step t.

Nonlinearity is introduced via a spike generation mechanism. The binary output $S^l[t]$ is generated by the Heaviside function $H(\cdot)$ when the accumulated potential $U^l[t]$ exceeds the firing threshold V_{th}. The mathematical expression is shown in Eq.(3):

$$S^l[t] = H(U^l[t] - V_{th}) = \begin{cases} 1, & U^l[t] \geq V_{th} \\ 0, & otherwise \end{cases} \tag{3}$$

After spiking, neurons reset their membrane potentials to prepare for the next firing. This process is illustrated in Eq.(4):

$$U^l[t] = \begin{cases} U^l[t] - V_{th} S^l[t], & soft \quad reset \\ U^l[t] \odot (1 - S^l[t]), & hard \quad reset \end{cases} \tag{4}$$

Here, $\odot$ denotes the element-wise multiplication.

3.2 SNN Training Framework

The training process relies on the Backpropagation Through Time (BPTT) algorithm [24]. To capture temporal dynamics, the gradients for the synaptic weights W^l are calculated by accumulating the derivatives of the loss function L across all time steps T:

$$\nabla_{W^l} L = \sum_{t=1}^{T} \frac{\partial L}{\partial U^l[t]} \frac{\partial U^l[t]}{\partial W^l} \tag{5}$$

To facilitate the backward flow of gradients through the non-differentiable spiking mechanism, the framework incorporates the SG strategy. In this work, the derivative of the spike generation function is approximated using a rectangular function $\mathbb{H}(\cdot)$:

$$\frac{\partial S^l[t]}{\partial U^l[t]} \approx \mathbb{H}(U^l[t]) = \frac{1}{\sigma} \cdot \mathbb{I}\left(\left|U^l[t] - V_{th}\right| < \frac{\sigma}{2}\right) \tag{6}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, and σ serves as a hyperparameter typically set to the threshold value V_{th}.

3.3 Voltage-gated Potassium Currents

Voltage-gated potassium channels are key intrinsic regulators of neuronal excitability. In particular, the A-type potassium current is recruited during the subthreshold depolarization phase. It generates an outward current that shunts excitatory inputs to delay spike initiation and regulate firing frequency [25]. After spiking, the delayed rectifier potassium channels dominate the repolarization phase. This current facilitates the restoration of the resting membrane potential and determines the depth of afterhyperpolarization [11]. Such coordinated dynamic regulation prevents stochastic firing and is critical for maintaining system stability [26].

4 Methodology

4.1 Limitation of LIF Neuron Model

For the LIF neuron model with soft reset, when a neuron fires at time step $t-1$ with a membrane potential $U[t-1]$ exceeding the threshold V_{th}, the residual potential $\delta[t-1]$ retained after the soft reset is calculated as:

$$\delta[t-1] = U[t-1] - V_{th} \tag{7}$$

where $\delta[t-1] > 0$. According to Eq.(2), the condition for generating a consecutive spike at time step t can be rewritten as:

$$\lambda\delta[t-1] + I[t] > V_{th} \tag{8}$$

Solving for the input current $I[t]$, we derive the minimum input current required to generate a spike, denoted as $I_{min}[t]$:

$$I_{min}[t] = V_{th} - \lambda\delta[t-1] \tag{9}$$

Given $0 < \lambda < 1$ and $\delta[t-1] > 0$, it strictly holds that $I_{min}[t] < V_{th}$. This inequality mathematically proves that the retention of $\delta[t-1]$ lowers the firing threshold, rendering the model more sensitive to subsequent inputs.

Furthermore, this inertia effect is exacerbated during continuous firing sequences. Consider a scenario where the neuron fires continuously over time steps $t = 1, \ldots, T$, driven by a constant input overload $\Delta I = I - V_{th} > 0$. The residual potential $\delta[T]$ accumulates recursively. By expanding the recursion $\delta[T] = \lambda\delta[T-1] + \Delta I$ from an initial state $\delta[0] = 0$, we obtain a geometric series:

$$\delta[T] = \Delta I \sum_{i=0}^{T-1} \lambda^i = \Delta I \frac{1-\lambda^T}{1-\lambda} \tag{10}$$

We now examine the minimum input required for the next spike at $t = T+1$, which is $I_{min}[T+1] = V_{th} - \lambda\delta[T]$. Since $\delta[T]$ increases monotonically with T, the minimum input current required for the next spike at $t = T+1$ decreases:

$$I_{min}[T+1] < I_{min}[T] \tag{11}$$

This demonstrates the cumulative inertia inherent in models with soft reset: continuous spiking monotonically reduces the threshold required to sustain firing. This phenomenon not only progressively diminishes the model's ability to discriminate different input intensities, but also increases its sensitivity to noise.

In the hard-reset LIF model, the membrane potential is forced to zero immediately after spiking. Although this prevents cumulative inertia, it also eliminates the residual potential, hindering the retention of historical activity information and resulting in a limited dynamic response range.

4.2 The Design of KvLIF Model

Inspired by the roles of potassium currents in regulating neuronal excitability and post-spike membrane dynamics, we propose KvLIF as a simplified, biologically inspired extension of the LIF model. As shown in Figure 2(c), an auxiliary adaptive state $K[t]$ integrates membrane-potential and spike-history information to jointly modulate subthreshold integration and post-spike reset. Here, $K[t]$ serves as a phenomenological abstraction of potassium-mediated feedback, rather than a direct representation of a physical potassium current or conductance.

Voltage- and Spike-History-Dependent Adaptive State. For layer l, the adaptive state is first updated using its previous value and the current membrane potential:

$$\hat{K}^l[t] = \alpha K^l[t-1] + \beta U^l[t] \tag{12}$$

Here, α controls the retention of the adaptive state from the previous time step, while β scales the contribution of the current membrane potential.

Potassium currents exhibit a surge during spiking, reminiscent of delayed rectifier or high-threshold activated potassium currents. This is incorporated by updating $K^l[t]$ with the spike output $S^l[t]$:

$$K^l[t] = \hat{K}^l[t] + S^l[t] \tag{13}$$

The linear coupling of $K^l[t]$ with $U^l[t]$ creates a negative feedback loop and serves as the short-term memory to adaptively modulate neuronal excitability before the membrane potential reaches the spike threshold. The coupling of $K^l[t]$ to $S^l[t]$ models the explosive opening of potassium channels during spiking, which serves two distinct functions:

1. **Rapid repolarization of the membrane potential**, inducing a refractory period immediately after a spike.
2. **Slow decay of hyperpolarization**, where elevated potassium conductance persists across time steps through the retained fraction α, making $K^l[t]$ a proxy for long-term spiking history in excitability regulation.

Shunting Synaptic Inputs. In biological neurons, local dendritic potassium channels can shunt postsynaptic responses and reduce excitatory drive. Inspired by this mechanism, we modulate the postsynaptic current $I^l[t]$ using the potassium channel state variable from the previous time step:

$$I^l[t] = I^l[t] \odot (1 - \gamma\sigma(K^l[t-1])) \tag{14}$$

Here, $\sigma(\cdot)$ denotes the sigmoid function, which maps the potassium current to a normalized activation probability, and γ scales the magnitude of the shunting effect. The term $(1 - \gamma \cdot \sigma(K^l[t-1]))$ functions as a dynamic shunting gate, decreasing the strength of input currents. This conductance-based gating mechanism regulates neuronal excitability during the subthreshold integration phase, slowing depolarization, preventing excessive spiking.

Potassium Channels for Adaptive Repolarization. To capture somatic repolarization dynamics, we implement an adaptive reset mechanism inspired by biological afterhyperpolarization. In the KvLIF model, the membrane potential is updated as:

$$U^l[t] = U^l[t] - S^l[t] \odot (V_{th} + \sigma(K^l[t])) \tag{15}$$

The reset depth here is dynamically modulated by $\sigma(K^l[t])$. Since $K^l[t]$ surges immediately upon spiking, this term forces the membrane potential to drop significantly below the resting membrane potential. Larger $K^l[t]$ values result in deeper resets, effectively extending the relative refractory period. This adaptive reset provides activity-dependent control of subsequent firing.

Building upon these foundational principles, we derive the KvLIF model as:

$$\begin{aligned} U^l[t] &= \lambda U^l[t-1] + I^l[t] \odot (1 - \gamma\sigma(K^l[t-1])) \\ S^l[t] &= H(U^l[t] - V_{th}) \\ K^l[t] &= \alpha K^l[t-1] + \beta U^l[t] + S^l[t] \\ U^l[t] &= U^l[t] - S^l[t] \odot (V_{th} + \sigma(K^l[t])) \end{aligned} \tag{16}$$

KvLIF coordinates subthreshold integration and post-spike reset through a shared adaptive state driven by membrane potential and spiking history (Algorithm 1, Appendix B). This formulation captures selected functional effects of potassium-mediated feedback while intentionally abstracting away detailed ion-channel biophysics to limit model complexity and retain the computational simplicity of the LIF framework. KvLIF primarily addresses intrinsic excitability regulation, whose implications for network-level homeostasis may depend on broader cellular and synaptic interactions [27, 28].

4.3 Dynamic Analysis of the KvLIF Model

We visualize the dynamic response of the proposed KvLIF neuron to demonstrate its self-adaptation capabilities. The visualization of the variation in the internal variable $K[t]$ is provided in Figure 6 (Appendix C). As shown in Figure 3, all models are subjected to identical Poisson input spike trains over 40 time steps.

Regulation of Neuronal Excitability. Both conventional LIF models exhibit high firing rates, indicating a general tendency toward over-excitability. In contrast, the KvLIF neuron regulates its firing to a lower rate of 17.50%, reflecting efficient self-adaptation that suppresses redundant spikes.

Post-spike Membrane Dynamics. In the soft-reset LIF, high post-spike residual potentials cause hypersensitivity to subsequent inputs (Figure 3, middle). The hard-reset LIF model forces the potential to the resting state, erasing the membrane potential history. The KvLIF neuron exhibits an afterhyperpolarization phase, where the potential drops significantly below the resting potential (Figure 3, right column). By incorporating the negative feedback mechanism, the KvLIF model inhibits excessive spiking, thereby enhancing the flexibility for encoding complex temporal patterns.

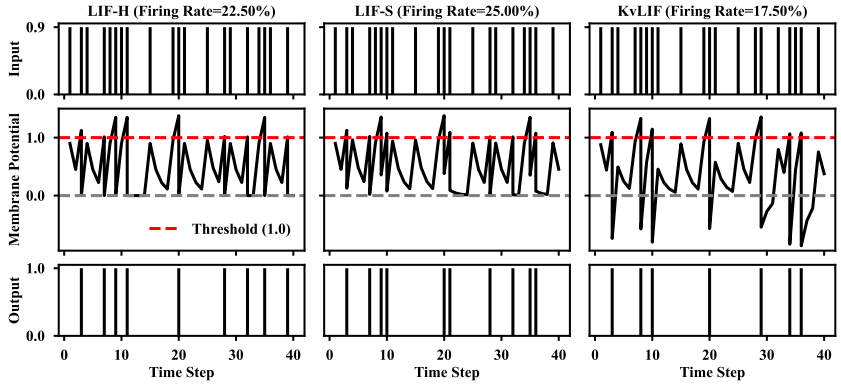
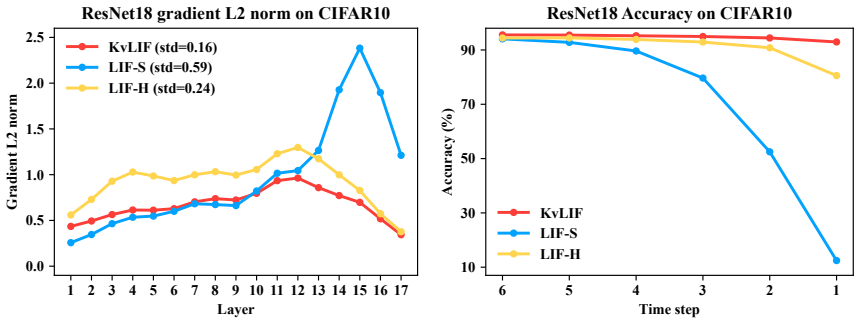


Figure 3: Comparison of dynamic responses between the KvLIF model and two conventional LIF models.



(a) Gradient Norm Distribution (b) Adaptability Analysis

Figure 4: Comparative performance of KvLIF and conventional LIF models. (a) Gradient L2 norm distribution. (b) Performance retention as inference time steps reduce.

4.4 Gradient Stability and Adaptability Analysis of the KvLIF model

The improved stability of KvLIF is shown in Figure 4(a). Layer-wise gradient L2 norms indicate that KvLIF enables more stable gradient propagation than conventional LIF models, as reflected by its lower standard deviation. Figure 4(b) further demonstrates its adaptability: although all models are trained with 6 time steps, KvLIF maintains over 90% accuracy when tested with only one time step as inference time steps are progressively reduced. This result suggests that KvLIF supports adaptive encoding and efficient inference within a short temporal window, which is further validated on neuromorphic datasets in Section 5.1.

Table 1: Performance comparison with existing methods on static image and neuromorphic datasets. The asterisk ($*$) denotes implementations achieved solely by modifying neuron models within the open source code [23]. [§]The CIFAR10-DVS test set contains 1,000 samples, so single-run accuracy changes in increments of 0.1 percentage points; the second decimal digit is therefore zero when percentages are reported to two decimal places.

Dataset	Method	Network Architecture	Time step	Accuracy (%)
CIFAR10	Dspike [29]	Modified ResNet-18	4 / 6	93.66 / 94.25
	GLIF [16]	ResNet-18	4 / 6	94.67 / 94.88
	SML [30]	ResNet-18	4 / 6	95.01 / 95.12
	CLIF [21]	ResNet-18	4 / 6	94.89 / 95.41
	ILIF [22]	ResNet-18	4 / 6	95.24 / 95.49
	KvLIF (Ours)	**ResNet-18**	**4 / 6**	**95.28 / 95.55**
	Temporal Pruning [31]	VGG-16	5	93.90
	ILIF [22]	VGG-16	6	94.25
	KvLIF (Ours)	**VGG-16**	**6**	**94.36**
CIFAR100	Dspike [29]	Modified ResNet-18	4 / 6	73.35 / 74.24
	GLIF [16]	ResNet-18	4 / 6	76.42 / 77.28
	SML [30]	ResNet-18	4 / 6	77.36 / 78.00
	CLIF [21]	ResNet-18	4 / 6	77.00 / 78.36
	ILIF [22]	ResNet-18	4 / 6	77.43 / 78.51
	KvLIF (Ours)	**ResNet-18**	**4 / 6**	**77.76 / 79.30**
	Temporal Pruning [31]	VGG-16	5	71.58
	ILIF [22]	VGG-16	6	75.25
	KvLIF (Ours)	**VGG-16**	**6**	**76.13**
Tiny ImageNet	ASGL [32]	VGG-13	4 / 8	56.57 / 56.81
	CLIF [21]	VGG-13	4 / 6	63.16 / 64.13
	CPT-SNN [33]	VGG-13	4 / 8	63.39 / 64.78
	KvLIF (Ours)	**VGG-13**	**4 / 6**	**64.03 / 64.82**
CIFAR10-DVS[§]	Dspike [29]	ResNet-18	10	75.40
	PLIF [14]	PLIF-Net	20	74.80
	GLIF [16]	7B-wideNet	16	78.10
	CLIF [21]	VGG-11	16	79.00
	ILIF [22]	VGG-11	10	78.60
	KvLIF (Ours)	**VGG-11**	**10**	**79.50**
	PSN [34]	VGGSNN	10	85.90
	CLIF [21]	VGGSNN	10	86.10
	AR-LIF [23]	VGGSNN	16	87.90
	KvLIF (Ours)*	**VGGSNN**	**10**	**88.40**
DVS-Gesture	PLIF [14]	PLIF-Net	20	97.57
	KLIF [35]	Modified PLIF Net	12	94.10
	CLIF [21]	VGG-11	20	97.92
	AGMM [36]	VGGSNN	16	97.92
	ILIF [22]	VGG-11	20	97.92
	KvLIF (Ours)	**VGG-11**	**10**	**98.26**

5 Experiments

We conduct a series of experiments to evaluate the proposed KvLIF neuron model, with all details provided in Appendix D.

5.1 Performance Comparison

We compare KvLIF with other competitive SNN methods, while detailed comparisons with conventional LIF models, SFA-based LIF models, and Artificial Neural Networks (ANNs) are provided in Appendices E and F. To isolate the effect of the proposed neuron, we use only standard preprocessing and exclude advanced augmentation strategies such as AutoAugment and CutMix.

As shown in Table 1, KvLIF achieves consistently strong performance on both static image and neuromorphic benchmarks across different network architectures. On static image datasets, it obtains 95.55% accuracy on CIFAR10 and 79.30% on CIFAR100 with Spiking ResNet-18 using six time steps, and 64.82% on Tiny ImageNet with Spiking VGG-13 using six time steps. Its advantage further extends to asynchronous spatiotemporal tasks: on CIFAR10-DVS [37], KvLIF achieves 79.50% accuracy with Spiking VGG-11 using ten time steps and 88.40% with VGGSNN, while reaching 98.26% on DVS-Gesture [38] with Spiking VGG-11 using ten time steps. We further evaluate KvLIF on three SNN Transformer architectures on CIFAR100, observing consistent accuracy gains over LIF (Appendix G). Additional runs with multiple random seeds show consistent performance trends for KvLIF (Appendix H). In most cases, KvLIF outperforms competing methods with fewer time steps, demonstrating its effectiveness and efficiency in modeling both spatial and spatiotemporal information. Moreover, we analyze the firing rates and energy consumption of KvLIF (Appendices I and J).

To evaluate temporal generalization, we further test KvLIF on the SHD neuromorphic audio dataset, where it outperforms SFA-based baselines and shows improved capability in capturing long-range temporal dependencies from sparse neuromorphic audio sequences (Appendix F).

Table 2: Robustness tests on CIFAR100 and DVS-Gesture under different noise levels.

	CIFAR100					DVS-Gesture				
Neuron	0.04	0.08	0.12	0.16	0.20	0.10	0.20	0.30	0.40	0.50
LIF-S	65.58	53.96	40.81	29.74	21.57	90.63	84.03	77.78	69.10	66.32
LIF-H	71.60	61.23	44.35	30.57	21.33	77.78	68.40	60.07	55.56	52.08
GLIF	69.71	62.38	50.40	38.19	28.05	79.86	79.86	77.08	67.36	58.68
CLIF	75.57	66.71	60.71	40.83	29.93	89.58	82.64	75.69	63.89	57.29
ILIF	74.52	65.33	50.24	36.49	25.44	86.46	76.74	70.83	61.46	56.25
KvLIF	**76.10**	**68.37**	**63.03**	**44.27**	**33.32**	**93.75**	**89.93**	**86.11**	**80.90**	**75.69**

Table 3: Ablation study of the proposed KvLIF neuron on different datasets.

Synaptic shunting	Adaptive repolarization	CIFAR10	CIFAR100	Tiny ImageNet	CIFAR10 -DVS	DVS-Gesture
✓	✓	95.55	79.30	64.82	79.50	98.26
✗	✓	95.46	79.22	64.23	78.60	97.57
✓	✗	94.26	77.43	62.82	77.60	97.22
✗	✗	94.09	77.05	62.70	77.20	96.18

5.2 Robustness Analysis

The adaptive regulation of neuronal excitability in KvLIF also enhances robustness to noise perturbations. For static image datasets, we use the Spiking ResNet-18 architecture and evaluate robustness by adding zero-mean Gaussian noise to the test set, with standard deviations ranging from 0.04 to 0.20. For neuromorphic datasets, we employ the Spiking VGG-11 architecture. To preserve data sparsity, noise is added to 10%-50% of randomly selected pixels in the test set. Results for CIFAR100 and DVS-Gesture are presented in Table 2, with additional robustness evaluations on other datasets provided in the Appendix (Tables 12 and 13, Appendix K).

Across both static image and neuromorphic experiments, the classification accuracy of all models decreases as noise intensity increases. Nevertheless, KvLIF consistently achieves the highest accuracy under all tested noise levels, indicating stronger robustness to spatial perturbations in static images and event noise in neuromorphic data streams. In addition, KvLIF remains robust to missing temporal events, further demonstrating that its adaptive regulation enhances resilience to corrupted or incomplete spatiotemporal inputs (Tables 14 and 15, Appendix L).

5.3 Ablation Study

We analyze the relative contributions of the two adaptive components in the KvLIF model: the A-type current, which integrates membrane potential to shunt synaptic inputs, and the delayed rectifier current, which accumulates spiking history to regulate post-spike reset dynamics. Detailed quantitative results are presented in Table 3. Adaptive repolarization plays a more significant role in the improved performance of KvLIF, with its ablation causing a larger performance drop across all tested datasets. In comparison, synaptic shunting has a smaller impact.

5.4 Hyperparameter Sensitivity Analysis

We analyze the sensitivity of KvLIF to its key hyperparameters, α, β, and γ. As shown in Table 4 (Appendix D), KvLIF requires limited per-dataset tuning: α and γ are fixed at 0.8 and 0.05 across all five benchmarks, while β only takes values of 0.1 or 0.3. Local sensitivity analysis around the default settings further shows that KvLIF remains stable under moderate hyperparameter perturbations. Moreover, γ is fixed at 0.05 because it is the least sensitive coefficient, consistent with the ablation results in Table 3, where removing synaptic shunting causes a smaller performance drop than removing adaptive repolarization. The full results are provided in Appendix M.

6 Conclusion

In this study, we address key limitations of conventional LIF models, namely their restricted response range and susceptibility to noise, which hinder adaptability in complex environments. To overcome these challenges, KvLIF introduces a conductance state $K[t]$ that integrates membrane-potential fluctuations and spiking history. Through this state, KvLIF dynamically modulates synaptic inputs before firing and adaptively adjusts the reset depth after firing. This coupled regulation of integration and reset expands the neuronal response range, suppresses redundant spikes, and improves temporal adaptability. Empirically, KvLIF achieves ANN-level accuracy on static benchmarks, suggesting that the proposed spiking neuron can narrow the conventional performance gap. Its advantages are more clearly reflected beyond accuracy: across static and neuromorphic datasets, KvLIF delivers stronger robustness, lower firing activity, and acceptable energy consumption. Overall, KvLIF highlights the value of potassium-inspired self-adaptation for building robust and efficient neuromorphic SNNs.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2023YFF1204200 to Y.Z.) and the National Natural Science Foundation of China (62476197 to Y.Z.).

References

[1] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[2] Wulfram Gerstner and Werner M Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[3] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

[4] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[5] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

[6] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

[7] Yufei Guo, Yuanpei Chen, Liwen Zhang, YingLei Wang, Xiaode Liu, Xinyi Tong, Yuanyuan Ou, Xuhui Huang, and Zhe Ma. Reducing information loss for spiking neural networks. In *European Conference on Computer Vision*, pages 36–52, 2022.

[8] Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

[9] Yunliang Zang, Eve Marder, and Shimon Marom. Sodium channel slow inactivation normalizes firing in axons with uneven conductance distributions. *Current biology*, 33(9):1818–1824, 2023.

[10] Jan Benda and Andreas VM Herz. A universal model for spike-frequency adaptation. *Neural computation*, 15(11):2523–2564, 2003.

[11] Bruce P Bean. The action potential in mammalian central neurons. *Nature Reviews Neuroscience*, 8(6): 451–465, 2007.

[12] Yunliang Zang, Sungho Hong, and Erik De Schutter. Firing rate-dependent phase responses of purkinje cells support transient oscillations. *Elife*, 9:e60692, 2020.

[13] Hartmut Fitz, Marvin Uhlmann, Dick Van den Broek, Renato Duarte, Peter Hagoort, and Karl Magnus Petersson. Neuronal spike-rate adaptation supports working memory in language processing. *Proceedings of the National Academy of Sciences*, 117(34):20881–20889, 2020.

[14] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2661–2671, 2021.

[15] Siqi Wang, Tee Hiang Cheng, and Meng-Hiot Lim. Ltmd: learning improvement of spiking neural networks with learnable thresholding neurons and moderate dropout. In *Advances in Neural Information Processing Systems*, volume 35, pages 28350–28362, 2022.

[16] Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pages 32160–32171, 2022.

[17] Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 16838–16847, 2024.

[18] Lihao Wang and Zhaofei Yu. Autaptic synaptic circuit enhances spatio-temporal predictive learning of spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 52083–52098, 2024.

[19] Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

[20] Maximilian Baronig, Romain Ferrand, Silvester Sabathiel, and Robert Legenstein. Advancing spatio-temporal processing through adaptation in spiking neural networks. *Nature Communications*, 16(1): 5776, 2025.

[21] Yulong Huang, Xiaopeng Lin, Hongwei Ren, Haotian Fu, Yue Zhou, Zunchang Liu, Biao Pan, and Bojun Cheng. Clif: Complementary leaky integrate-and-fire neuron for spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 19949–19972, 2024.

[22] Kai Sun, Peibo Duan, Levin Kuhlmann, Beilun Wang, and Bin Zhang. Ilif: Temporal inhibitory leaky integrate-and-fire neuron for overactivation in spiking neural networks. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 4200–4208, 2025.

[23] Zeyu Huang, Wei Meng, Quan Liu, Kun Chen, and Li Ma. Ar-lif: Adaptive reset leaky integrate-and-fire neuron for spiking neural networks. *arXiv preprint arXiv:2507.20746*, 2025.

[24] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[25] JA Connor and CF Stevens. Prediction of repetitive firing behaviour from voltage clamp data on an isolated neurone soma. *The Journal of physiology*, 213(1):31-53, 1971.

[26] Helene Vacher, Durga P Mohapatra, and James S Trimmer. Localization and targeting of voltage-dependent ion channels in mammalian central neurons. *Physiological reviews*, 88(4):1407–1447, 2008.

[27] Eve Marder and Jean-Marc Goaillard. Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*, 7(7):563–574, 2006.

[28] Timothy O'Leary, Alex H Williams, Alessio Franci, and Eve Marder. Cell types, network homeostasis, and pathological compensation from a biologically plausible ion channel expression model. *Neuron*, 82 (4):809–821, 2014.

[29] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. In *Advances in neural information processing systems*, volume 34, pages 23426–23439, 2021.

[30] Shikuang Deng, Hao Lin, Yuhang Li, and Shi Gu. Surrogate module learning: Reduce the gradient error accumulation in training spiking neural networks. In *International Conference on Machine Learning*, pages 7645–7657, 2023.

[31] Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning. In *European Conference on Computer Vision*, pages 709–726, 2022.

[32] Ziming Wang, Runhao Jiang, Shuang Lian, Rui Yan, and Huajin Tang. Adaptive smoothing gradient learning for spiking neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 35798–35816, 2023.

[33] Qisheng Xia, Yang Yu, Zheng Chang, Bin Hui, and Haibo Luo. CPT-SNN: A spiking neural network that can combine the previous timestep. *Neurocomputing*, 640:130253, 2025.

[34] Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. In *Advances in Neural Information Processing Systems*, volume 36, pages 53674–53687, 2023.

[35] Chunming Jiang and Yilei Zhang. Klif: An optimized spiking neuron unit for tuning surrogate gradient function. *Neural Computation*, 36(12):2636–2650, 2024.

[36] Yu Liang, Wenjie Wei, Ammar Belatreche, Honglin Cao, Zijian Zhou, Shuai Wang, Malu Zhang, and Yang Yang. Towards accurate binary spiking neural networks: Learning with adaptive gradient modulation mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1402–1410, 2025.

[37] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

[38] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017.

[39] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022.

[40] Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3174–3182, 2023.

[41] Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023.

[42] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022.

[43] Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In *Advances in Neural Information Processing Systems*, volume 36, pages 64043–64058, 2023.

[44] Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using Q-K attention. In *Advances in Neural Information Processing Systems*, volume 37, pages 13074–13098, 2024.

[45] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[46] Edgar Lemaire, Loïc Cordone, Andrea Castagnetti, Pierre-Emmanuel Novac, Jonathan Courtois, and Benoît Miramond. An analytical estimation of spiking neural networks energy efficiency. In *Neural Information Processing (ICONIP 2022)*, volume 13623 of *Lecture Notes in Computer Science*, pages 574–587. Springer, 2023.

[47] Hui Chen, Lin Jiang, Yuanyong Luo, Zhonghai Lu, Yuxiang Fu, Li Li, and Zongguang Yu. A CORDIC-based architecture with adjustable precision and flexible scalability to implement sigmoid and tanh functions. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020.

A Visualization of Feature Maps

We visualize the feature maps of the encoding layer in the LIF-based ResNet-18 on CIFAR10, as shown in Figure 5. Observations reveal that the input intensities received by LIF neurons significantly exceed 1.0, confirming that these neurons indeed process high-intensity signals.

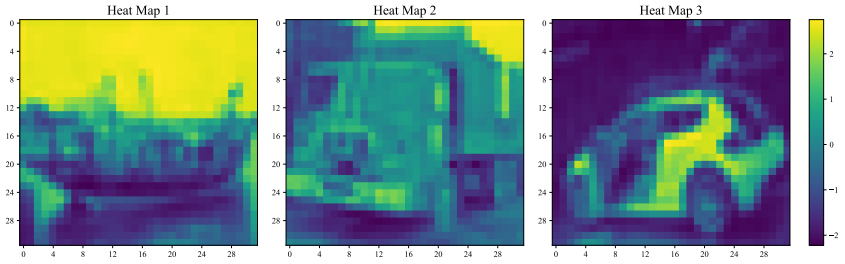


Figure 5: Heatmap visualization of the encoding layer responses to different input samples.

B Core computation of the KvLIF model

The pseudo-code for KvLIF (Algorithm 1) outlines the sequence of operations, illustrating how the potassium current state variable interacts with the membrane potential and preceding spiking history.

Functioning as a dual-phase regulator, $K^l[t]$ modulates subthreshold depolarization through A-type potassium current-like mechanisms and controls repolarization depth via delayed rectifier potassium current-like mechanisms. This integrated regulation ensures that the integration and reset phases remain precisely coordinated at every time step.

C Visualization of K variable

Under a constant input current of 0.8, we visualize the temporal variation of $K[t]$ over 100 time steps. The results show that $K[t]$ remains dynamically modulated throughout the simulation, rather than converging to a static state (Figure 6). This observation indicates that the dynamic behavior of $K[t]$ retains activity-dependent regulatory properties.

D Detailed Experimental Settings and Dataset Preprocessing

Experiments on all datasets except Tiny ImageNet are conducted on a single NVIDIA RTX 4090 GPU, while experiments on Tiny ImageNet are conducted on a single NVIDIA A800 GPU. We employ the Rectangle surrogate gradient function with $\sigma = V_{th} = 1$ and set the decay constant to 0.5. The random seed is fixed at 1000. For the loss function, we utilize the TET method with a lambda value of 0.05 [39]. Detailed training hyperparameters are listed in Table 4.

Algorithm 1 Core computation of the KvLIF model

Input: Input current I, time step t, previous membrane potential U_{pre}, previous state variable K_{pre}.
Hyperparameters: Decay constant λ, threshold V_{th}, Potassium current decay factor α, membrane potential scaling factor β, shunting effect scaling factor γ.
Output: Spike S

if $t = 0$ **then**
  Set $U_{pre} \leftarrow 0$ and $K_{pre} \leftarrow 0$ ▷ initialization
end if
$I \leftarrow I \odot (1 - \gamma\sigma(K_{pre}))$ ▷ Shunting synaptic inputs
$U \leftarrow \lambda U_{pre} + I$ ▷ leaky & integrate
$K_{init} \leftarrow \alpha K_{pre} + \beta U$ ▷ Potassium currents
$S \leftarrow H(U - V_{th})$ ▷ fire
$K \leftarrow K_{init} + S$ ▷ Potassium currents surge
$U \leftarrow U - S \odot V_{th}$ ▷ reset
$U \leftarrow U - S \odot \sigma(K)$ ▷ Adaptive repolarization.

Return S

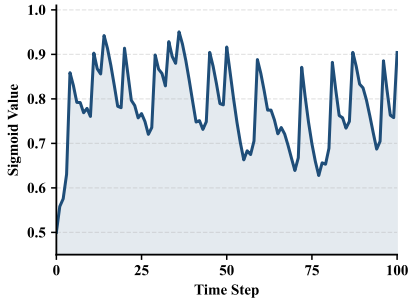


Figure 6: Temporal dynamics of the K variable.

CIFAR10 and CIFAR100. These datasets each contain 60,000 color images (32 × 32 pixels), divided into 50,000 training and 10,000 testing samples across 10 and 100 classes, respectively. Data preprocessing includes random cropping with 4-pixel padding, random horizontal flipping, and Cutout. Image data are normalized to zero mean and unit variance. We employ the direct encoding method [40], where pixel values are repeatedly input into the model at each time step. The backbone network for these tasks is Spiking ResNet-18.

Tiny ImageNet. This dataset contains 100,000 color images (64 × 64 pixels) from 200 classes. Our preprocessing pipeline includes random cropping with 8-pixel padding and horizontal flipping. Direct encoding is utilized for input representation. The backbone network employed is Spiking VGG-13.

CIFAR10-DVS. This neuromorphic dataset comprises 10,000 event-driven samples derived from CIFAR-10, obtained by recording the original images using a Dynamic Vision Sensor (DVS) [37]. Each sample features an original spatial resolution of 128 × 128 pixels. Each sample is processed using event-to-frame integration, with the spatial resolution at each time step resized to 48 × 48.

Table 4: Training hyperparameter settings for various datasets.

Dataset	Batch size	Epoch	Learning rate	Optimizer	Weight decay	α	β	γ
CIFAR10	128	200	0.1	SGD	5e-5	0.8	0.3	0.05
CIFAR100	128	300	0.1	SGD	5e-4	0.8	0.1	0.05
Tiny ImageNet	256	300	0.1	SGD	5e-4	0.8	0.3	0.05
CIFAR10-DVS	128	200	0.05	SGD	5e-4	0.8	0.1	0.05
DVS-Gesture	16	150	5e-4	Adam	0.0	0.8	0.3	0.05

This is followed by random cropping with 4-pixel padding to generate the final preprocessed samples. We adopt Spiking VGG-11 as the backbone network.

DVS-Gesture. This dataset captures 11 distinct hand gesture categories performed by 29 subjects under three different lighting conditions [38]. For preprocessing, we utilize the SpikingJelly framework to implement event-to-frame integration [41]. The backbone network employed is Spiking VGG-11.

E Performance Comparison with Conventional LIF models and ANNs

To evaluate the effectiveness of the proposed KvLIF neuron, we compare it with conventional LIF models on both static and neuromorphic datasets, and further include ANN counterparts with ReLU activations on CIFAR10 and CIFAR100, as summarized in Table 5. The experiments cover multiple network architectures and time steps, allowing us to assess whether KvLIF improves spiking dynamics while narrowing the performance gap between SNNs and ANNs.

Experimental results show that KvLIF consistently outperforms conventional LIF-S and LIF-H baselines across all evaluated datasets and architectures. On static image datasets, KvLIF achieves strong accuracy at both $T = 4$ and $T = 6$. Notably, on CIFAR10, KvLIF reaches 95.55% at $T = 6$, approaching the ReLU ANN counterpart with only a 0.16% gap. On CIFAR100, KvLIF obtains 79.30% at $T = 6$, surpassing the ReLU ANN baseline by 0.99%. On Tiny ImageNet, KvLIF also achieves the best performance among spiking models, reaching 64.82% at $T = 6$. For neuromorphic tasks, KvLIF demonstrates superior temporal processing ability, attaining 79.50% on CIFAR10-DVS and 98.26% on DVS-Gesture. These results indicate that the adaptive potassium-regulated dynamics not only improve over conventional LIF neurons, but also enable competitive or superior performance compared with ANN counterparts on static benchmarks where such comparisons are available.

Table 5: Performance comparison with conventional LIF models.

Dataset	Architecture	Neuron	Time step	Accuracy (%)
CIFAR10	ResNet-18	ReLU	1	95.71
	ResNet-18	LIF-S	4 / 6	91.54 / 94.09
	ResNet-18	LIF-H	4 / 6	94.42 / 94.70
	ResNet-18	**KvLIF**	**4 / 6**	**95.28 / 95.55**
CIFAR100	ResNet-18	ReLU	1	78.31
	ResNet-18	LIF-S	4 / 6	69.24 / 77.05
	ResNet-18	LIF-H	4 / 6	74.64 / 77.17
	ResNet-18	**KvLIF**	**4 / 6**	**77.76 / 79.30**
Tiny ImageNet	VGG-13	ReLU	1	62.86
	VGG-13	LIF-S	4 / 6	60.72 / 62.70
	VGG-13	LIF-H	4 / 6	61.58 / 61.81
	VGG-13	**KvLIF**	**4 / 6**	**64.03 / 64.82**
CIFAR10-DVS	VGG-11	LIF-S	10	77.20
	VGG-11	LIF-H	10	77.90
	VGG-11	**KvLIF**	**10**	**79.50**
DVS-Gesture	VGG-11	LIF-S	10	96.18
	VGG-11	LIF-H	10	96.88
	VGG-11	**KvLIF**	**10**	**98.26**

F Performance Comparison with LIF models incorporating SFA

KvLIF is compared with ALIF and adLIF across static image, neuromorphic vision, and neuromorphic audio datasets under different architectures and time steps. As shown in Table 6, KvLIF consistently outperforms the baseline models on both static image and neuromorphic vision tasks, demonstrating its advantage in handling different visual input modalities.

The comparison is further extended to the SHD neuromorphic audio dataset in Table 7. Under recurrent architectures with time steps no fewer than 250, KvLIF obtains $94.78 \pm 0.54\%$ with a

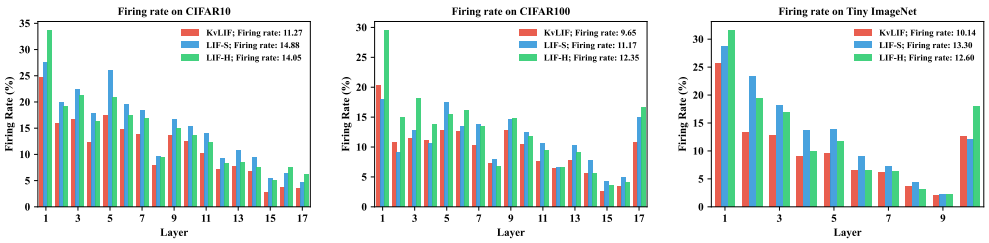
one-layer RNN and $95.98 \pm 0.11\%$ with a two-layer RNN, both surpassing the corresponding adLIF models. These results indicate that KvLIF provides more effective temporal feature representation and exhibits strong generalization ability across different input modalities and network structures.

Table 6: Performance comparison with LIF models incorporating SFA.

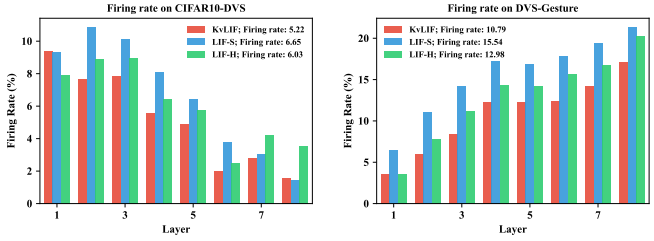
Dataset	Architecture	Neuron	Time step	Accuracy (%)
CIFAR10	ResNet-18	ALIF	4	94.44
	ResNet-18	adLIF	4	94.41
	ResNet-18	**KvLIF**	**4**	**95.28**
CIFAR100	ResNet-18	ALIF	4	75.75
	ResNet-18	adLIF	4	72.99
	ResNet-18	**KvLIF**	**4**	**77.76**
CIFAR10-DVS	VGG-11	ALIF	10	78.40
	VGG-11	adLIF	10	77.70
	VGG-11	**KvLIF**	**10**	**79.50**
DVS-Gesture	VGG-11	ALIF	10	97.22
	VGG-11	adLIF	10	97.92
	VGG-11	**KvLIF**	**10**	**98.26**

Table 7: Performance comparison on a neuromorphic audio dataset.

Dataset	Architecture	Neuron	Time step	Accuracy (%)
SHD	RNN	ALIF	250	90.40
	RNN (1 layer)	adLIF	250	94.59 ± 0.27
	RNN (2 layers)	adLIF	250	95.81 ± 0.56
	RNN (1 layer)	**KvLIF**	**250**	$\mathbf{94.78 \pm 0.54}$
	RNN (2 layers)	**KvLIF**	**250**	$\mathbf{95.98 \pm 0.11}$



(a) CIFAR10. (b) CIFAR100. (c) Tiny ImageNet.



(d) CIFAR10-DVS (e) DVS-Gesture.

Figure 7: Firing rate comparison across different datasets.

G Performance Evaluation on Transformer Architectures

To demonstrate the broad applicability of KvLIF, we evaluated its performance on three SNN Transformer architectures, including Spikformer [42], Spike-Driven Transformer [43], and QKFormer [44], using the CIFAR100 benchmark. As shown in Table 8, under identical training configurations and time steps, simply replacing standard LIF with KvLIF consistently yields accuracy gains across all Transformer backbones.

Table 8: Performance comparison on CIFAR100 with SNN Transformer architectures.

Architecture	Neuron	Accuracy (%)
Spikformer	LIF	77.68
	KvLIF	**77.95**
Spike-Driven Transformer	LIF	78.57
	KvLIF	**78.82**
QKFormer	LIF	80.57
	KvLIF	**81.43**

H Multi-seed evaluation across all datasets

To further evaluate the reliability of the proposed KvLIF model, we conduct multi-seed experiments across all datasets. The results are reported as the mean and standard deviation over three random seeds. As shown in Table 9, KvLIF consistently maintains strong performance across different datasets, architectures, and input modalities. The small performance variations indicate that the improvements brought by KvLIF are not dependent on a specific random initialization. These results further confirm the consistent performance across the evaluated random initializations.

Table 9: Multi-seed results across all datasets.

Dataset	Architecture	Neuron	Time step	Accuracy (%)
CIFAR10	ResNet-18	KvLIF	4	95.23 ± 0.04
CIFAR100	ResNet-18	KvLIF	4	77.85 ± 0.15
Tiny ImageNet	VGG-13	KvLIF	4	64.07 ± 0.06
CIFAR10-DVS	VGG-11	KvLIF	10	79.60 ± 0.14
DVS-Gesture	VGG-11	KvLIF	10	98.26 ± 0.00

I Firing Rate Analysis

We calculate the layer-wise firing rates of KvLIF compared to conventional LIF models across five datasets, as shown in Figure 7. The results demonstrate that KvLIF consistently achieves lower firing rates on both static image datasets (CIFAR10, CIFAR100, Tiny ImageNet) and neuromorphic datasets (CIFAR10-DVS, DVS-Gesture). This significant reduction in spiking activity is attributed to the intrinsic adaptive mechanism of the KvLIF neuron. By effectively suppressing the generation of redundant spikes, KvLIF ensures that only task-relevant information is transmitted, thereby maximizing computational sparsity.

J Energy Consumption Analysis

To evaluate the energy efficiency of the proposed model, we analyze the total energy consumption based on Synaptic Operations (SOP) [43]. The total energy consumption is calculated by summing the accumulation (AC) and multiply-accumulate (MAC) operations, weighted by their respective unit energy costs. The formulation is given by:

$$SOP = E_{AC} \cdot N_{AC} + E_{MAC} \cdot N_{MAC}, \tag{17}$$

where E_{AC} and E_{MAC} denote the energy cost per operation, and N_{AC} and N_{MAC} represent the total number of AC and MAC operations, respectively. Following established energy models in

neuromorphic computing, we set E_{AC} to 0.9 pJ for one 32-bit floating-point AC and E_{MAC} to 4.6 pJ for one 32-bit floating-point MAC operation [45].

This efficiency analysis is based on AC/MAC accounting and provides a hardware-agnostic reference for computational complexity, rather than a precise estimate of deployment cost. In particular, it does not include the overhead of sigmoid nonlinearities or the storage and memory read/write costs introduced by the additional potassium state variable. Thus, the reported complexity should be viewed as a basic comparison across models rather than a complete measure of runtime overhead on specific hardware platforms.

In our implementation, the total cost of each layer l is divided into two parts: (1) spike-driven synaptic accumulations and (2) neuron-state updates. We define $C^{(l)}_{\text{dense}}$ as the ANN-equivalent dense operation count for layer l, i.e., $C_{\text{out}}H_{\text{out}}W_{\text{out}}C_{\text{in}}k_hk_w$ for a convolutional layer, and $N_{\text{in}}N_{\text{out}}$ for a fully connected layer. For SNNs, this dense computation is converted into synaptic ACs according to

$$N^{(l)}_{\text{AC}} = C^{(l)}_{\text{dense}} \cdot fr^{(l)}_{\text{in}} \cdot T, \tag{18}$$

where $fr^{(l)}_{\text{in}}$ is the input firing rate of layer l and T is the simulation length. The MAC term is used separately for neuron-state updates. KvLIF has a larger neuron-level MAC term because, beyond standard membrane integration and reset, it additionally updates the potassium state, applies shunting modulation to the input current, and performs adaptive repolarization.

The network-level N_{AC} and N_{MAC} are obtained by summing the corresponding terms over all layers. As shown in Table 10, although KvLIF introduces additional local adaptive computations for potassium current dynamics and thus increases the neuron-level MAC cost, its lower firing rate reduces spike-driven synaptic ACs across layers. This reduction effectively offsets the extra computation, allowing KvLIF to maintain comparable or even lower overall energy consumption than conventional LIF models. These results indicate that incorporating potassium channel mechanisms can improve the accuracy of SNNs while preserving their energy efficiency.

Table 10: The energy consumption of synaptic operation on different datasets.

Dataset	Architecture	Neuron	T	ACs (M)	MACs (M)	SOP Energy (μJ)
CIFAR10	ResNet-18	ReLU	1	0.00	549.13	2526.00
	ResNet-18	LIF-S	6	512.82	3.34	476.91
	ResNet-18	LIF-H	6	487.93	3.34	454.51
	ResNet-18	KvLIF	6	393.16	13.37	415.34
CIFAR100	ResNet-18	ReLU	1	0.00	549.18	2526.22
	ResNet-18	LIF-S	6	359.34	3.34	338.78
	ResNet-18	LIF-H	6	404.70	3.34	379.61
	ResNet-18	KvLIF	6	320.60	13.37	350.05
Tiny ImageNet	VGG-13	ReLU	1	0.00	913.46	4201.90
	VGG-13	LIF-S	6	885.85	6.00	824.86
	VGG-13	LIF-H	6	822.80	6.00	768.11
	VGG-13	KvLIF	6	679.14	23.99	721.58
DVS-Gesture	VGG-11	LIF-S	10	3674.48	24.25	3418.58
	VGG-11	LIF-H	10	2992.54	24.25	2804.83
	VGG-11	KvLIF	10	2503.10	96.99	2698.96
CIFAR10-DVS	VGG-11	LIF-S	10	302.70	3.41	288.15
	VGG-11	LIF-H	10	264.66	3.41	253.88
	VGG-11	KvLIF	10	241.49	13.64	280.12

To account for costs beyond the operation-based estimate above, we further include memory reads and writes, synaptic and neuronal operations, and data addressing, following prior work [21, 46], with an estimated cost of 12.35 pJ per sigmoid evaluation [47]. Although the broader accounting in Table 11 indicates higher total energy consumption for KvLIF than for LIF, this additional cost accompanies consistent gains in accuracy and robustness. Moreover, KvLIF retains a substantial energy advantage over the corresponding ReLU ANN on static datasets, supporting a favorable performance–energy trade-off.

Table 11: Extended energy estimates including memory reads and writes, synaptic and neuronal computation, and data addressing.

Dataset	Neuron	Mem. Read & Write (mJ)	Synaptic & Neuron Op. (mJ)	Addr. (mJ)	Total (mJ)
CIFAR10	ReLU	109.9045	1.7573	0.0001	111.6619
	LIF	34.8653	0.0190	0.0123	34.8966
	KvLIF	61.5563	0.1211	0.0185	61.6960
CIFAR100	ReLU	109.9091	1.7574	0.0001	111.6667
	LIF	25.3054	0.0170	0.0088	25.3312
	KvLIF	50.9449	0.1203	0.0153	51.0804
Tiny ImageNet	ReLU	182.8268	2.9232	0.0002	185.7501
	LIF	44.5320	0.0446	0.0157	44.5923
	KvLIF	79.8119	0.2282	0.0241	80.0642
CIFAR10-DVS	LIF	9.1751	0.0234	0.0031	9.2016
	KvLIF	17.1263	0.1295	0.0050	17.2608
DVS-Gesture	LIF	110.8881	0.1168	0.0377	111.0425
	KvLIF	176.7999	0.8593	0.0514	177.7106

K Additional Robustness Test

We conduct additional robustness tests on the static image dataset (CIFAR-10) and the neuromorphic dataset (CIFAR10-DVS). The noise settings follow the protocols described in Section 5.2 of the main text, and the results are summarized in Tables 12 and 13.

In the static image experiments, accuracy naturally degrades for all models as noise intensity increases. The KvLIF model consistently achieves higher accuracy than other neuron models across almost all noise levels. The only exception is at the 0.08 noise level, where it is slightly surpassed by others. Similarly, in the neuromorphic experiments, while all models exhibit reduced accuracy with increasing noise proportions, KvLIF maintains the highest accuracy at all levels, demonstrating superior resilience to event noise.

Table 12: Robustness test on CIFAR10 under different noise levels.

Neuron	0.04	0.08	0.12	0.16	0.20
LIF-S	90.65	86.74	79.91	70.96	60.75
LIF-H	93.72	90.87	86.38	79.59	71.82
GLIF	93.64	91.48	87.92	82.05	74.97
CLIF	94.44	92.46	88.40	82.69	74.98
ILIF	94.31	**92.96**	88.77	83.63	76.61
KvLIF	**94.84**	92.73	**89.12**	**84.12**	**77.02**

Table 13: Robustness test on CIFAR10-DVS under different noise levels.

Neuron	0.10	0.20	0.30	0.40	0.50
LIF-S	75.20	73.80	69.30	65.30	58.80
LIF-H	75.90	74.70	69.20	66.30	61.20
GLIF	70.20	61.40	51.00	40.60	31.30
CLIF	75.40	73.20	69.60	66.10	60.30
ILIF	75.80	75.70	72.80	70.00	64.40
KvLIF	**77.10**	**75.90**	**74.40**	**72.10**	**69.00**

L Temporal Robustness Analysis

To verify the temporal robustness of KvLIF, we introduce temporal noise by randomly masking out a certain percentage of event time steps in the test sets. These experiments are conducted on both the DVS-Gesture and CIFAR10-DVS datasets, with drop rates ranging from 0.1 to 0.5.

Table 14: Temporal Robustness test on DVS-Gesture under different noise levels.

Neuron	0.10	0.20	0.30	0.40	0.50
LIF-S	96.53	95.83	94.10	92.36	89.93
LIF-H	95.83	95.14	90.28	78.47	60.07
GLIF	95.83	96.18	96.18	94.79	93.06
CLIF	96.18	95.49	94.44	93.06	90.97
ILIF	96.86	96.53	94.44	92.01	88.54
KvLIF	**97.22**	**96.53**	**96.53**	**95.83**	**94.10**

While the accuracy of all models naturally declines as noise levels increase, the KvLIF model demonstrates stronger temporal robustness. Specifically, on the DVS-Gesture dataset, KvLIF maintains the highest accuracy across all drop rates (Table 14). Similarly, on CIFAR10-DVS, it outperforms competing models at all noise levels, with the sole exception of the 0.4 drop rate (Table 15).

Table 15: Temporal Robustness test on CIFAR10-DVS under different noise levels.

Neuron	0.10	0.20	0.30	0.40	0.50
LIF-S	75.40	74.10	71.50	66.60	55.10
LIF-H	77.40	76.00	74.00	71.40	61.60
GLIF	74.10	69.70	63.80	55.50	43.70
CLIF	77.90	76.30	74.10	68.50	57.70
ILIF	78.10	76.20	74.60	**71.60**	63.80
KvLIF	**79.00**	**77.90**	**74.70**	71.10	**65.00**

M Hyperparameter Sensitivity Analysis Results

To further evaluate the robustness of KvLIF to hyperparameter choices, we conduct a local sensitivity analysis around the default settings of α, β, and γ. Specifically, we vary α and β within a moderate range while keeping γ fixed at 0.05, and additionally test different values of γ across multiple datasets (Tables 16, 17, 18, 19, and 20). The results show that KvLIF maintains stable performance on CIFAR-10, CIFAR10-DVS, and DVS-Gesture under moderate changes of α and β, indicating that the model is not highly dependent on precise hyperparameter tuning. CIFAR-100 exhibits relatively higher sensitivity, especially with respect to β, but the performance remains competitive across a broad range of parameter settings.

For γ, the results are consistently stable across all tested datasets (Table 20), supporting our choice of fixing γ=0.05 in the main experiments. Overall, these results demonstrate that KvLIF achieves robust performance with only mild adjustment of β for datasets with different input statistics.

Table 16: Hyperparameter sensitivity analysis on CIFAR10 (γ is fixed at 0.05)

$\alpha\backslash\beta$	0.2	0.3	0.4
0.7	95.26%	95.00%	95.13%
0.8	95.12%	95.28%	95.03%
0.9	95.21%	95.07%	95.13%

Table 17: Hyperparameter sensitivity analysis on CIFAR100 (γ is fixed at 0.05)

$\alpha\backslash\beta$	0.2	0.3	0.4
0.7	77.33%	77.06%	77.02%
0.8	78.03%	77.76%	76.75%
0.9	77.66%	75.46%	77.43%

Table 18: Hyperparameter sensitivity analysis on CIFAR10-DVS (γ is fixed at 0.05)

$\alpha\backslash\beta$	0.2	0.3	0.4
0.7	78.70%	79.20%	78.90%
0.8	78.90%	79.50%	79.10%
0.9	79.00%	78.70%	78.60%

Table 19: Hyperparameter sensitivity analysis on DVS-Gesture (γ is fixed at 0.05)

$\alpha\backslash\beta$	0.2	0.3	0.4
0.7	97.92%	97.57%	97.92%
0.8	97.22%	98.26%	97.57%
0.9	97.57%	97.57%	97.57%

Table 20: Hyperparameter sensitivity analysis results for γ.

Dataset	0.01	0.05	0.1
CIFAR10	94.96%	95.28%	95.07%
CIFAR100	77.63%	77.76%	76.19%
CIFAR10-DVS	78.90%	79.50%	78.90%
DVS-Gesture	97.92%	98.26%	97.57%

N Limitations and Future Work

While KvLIF demonstrates improvements in accuracy, robustness, and temporal representation, several limitations remain. First, our energy analysis includes both AC/MAC-based estimates and extended accounting for memory reads and writes, data addressing, and nonlinear computations (Appendix J). The extended estimates indicate that KvLIF incurs higher total energy consumption than the evaluated LIF baseline, despite its reduced spiking activity, while remaining below the corresponding ReLU ANNs on static datasets. These results highlight a trade-off between improved performance and the overhead of adaptive neuronal dynamics. However, these estimates rely on assumed operation and memory-access costs and have not been validated through measurements on neuromorphic hardware. Future work should evaluate energy consumption and latency on specific platforms and investigate efficient state storage and nonlinear-function implementations to reduce the overhead of potassium-mediated adaptation.

Second, our evaluation focuses on classification tasks involving static images, neuromorphic vision, and neuromorphic audio. Although these benchmarks support the applicability of KvLIF across multiple input modalities and network architectures, its effectiveness beyond classification remains unverified. Future work could examine temporal detection, event-based tracking, and sequential decision-making to assess whether the proposed adaptive dynamics provide similar benefits under continuous inputs and task-specific temporal constraints.

O Impact Statement

This work bridges biophysical principles and efficient model design, demonstrating how biologically inspired self-regulation can create more robust and adaptable SNNs. By improving noise resilience

and dynamic range, it directly contributes to the development of reliable and energy-efficient neuromorphic computing for real-world applications like edge AI and autonomous systems.

P Code Availability

The code for the KvLIF neuron model and the accompanying experiments is publicly available at https://github.com/ZubinYang/KvLIF.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have summarized the main contributions and scope in the abstract and introduction sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We point out the limitations in Appendix N.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We present the assumptions and provide a complete proof in Section 4

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have listed the training details in Appendix D, from which people can reproduce the main experimental results based on it. Our code is publicly available on GitHub.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide code in the supplementary materials. The open-source datasets are listed in Appendix D.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We discuss the experimental setup and training details in Section 5 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the multi-seed evaluation across all datasets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We summarize our computational resources in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the Code of Ethics and make sure to preserve anonymity.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed potential societal impacts in Appendix O.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have used open-source code and datasets, and have appropriately cited them.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This work does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.