IML-Spikeformer: Input-aware Multi-Level Spiking Transformer for Speech Processing

Zeyang Song, Shimin Zhang, Yuhong Chou, Jibin Wu[†], *Member, IEEE*, Haizhou Li[†], *Fellow, IEEE*

Abstract—Spiking Neural Networks (SNNs), inspired by biological neural mechanisms, represent a promising neuromorphic computing paradigm that offers energy-efficient alternatives to traditional Artificial Neural Networks (ANNs). Despite proven effectiveness, SNN architectures have struggled to achieve competitive performance on large-scale speech processing tasks. Two key challenges hinder progress: (1) the high computational overhead during training caused by multi-timestep spike firing, and (2) the absence of large-scale SNN architectures tailored to speech processing tasks. To overcome the issues, we introduce Input-aware Multi-Level Spikeformer, i.e. IML-Spikeformer, a spiking Transformer architecture specifically designed for large-scale speech processing. Central to our design is the Input-aware Multi-Level Spike (IMLS) mechanism, which simulates multi-timestep spike firing within a single timestep using an adaptive, input-aware thresholding scheme. IML-Spikeformer further integrates a Re-parameterized Spiking Self-Attention (RepSSA) module with a Hierarchical Decay Mask (HDM), forming the HD-RepSSA module. This module enhances the precision of attention maps and enables modeling of multi-scale temporal dependencies in speech signals. Experiments demonstrate that IML-Spikeformer achieves word error rates of 6.0% on AiShell-1 and 3.4% on Librispeech-960, comparable to conventional ANN transformers while reducing theoretical inference energy consumption by 4.64× and 4.32× respectively. IML-Spikeformer marks an advance of scalable SNN architectures for large-scale speech processing in both task performance and energy efficiency. Our source code and model checkpoints are publicly available at github.com/Pooookeman/IML-Spikeformer

Index Terms—Spiking Neural Networks, Neuromorphic Auditory Processing, Speech Recognition, Spiking Transformer

I. INTRODUCTION

RECENT advances in speech processing benefit from large-scale deep learning models, particularly Transformer-based architectures [1]. The high computational cost of such models has motivated the search for energy-efficient alternatives. Spiking Neural Networks (SNNs), which mimic the event-driven information processing of biological neurons, have emerged as a promising solution. Unlike traditional artificial neural networks (ANNs), SNNs emulate the dynamics of biological neurons and utilize spike trains for the representation of information [2]. The event-driven nature of Spiking Neural Networks (SNNs) enables asynchronous computation on neuromorphic hardware [3]–[5], where a neuron responds only upon arrival of a spike. This sparse activation mechanism significantly reduces power consumption, making SNNs particularly suitable for energy-constrained edge computing applications [6], [7].

Beyond their energy efficiency advantages, SNNs have demonstrated remarkable performance across diverse application domains. Substantive empirical evidence validates their effectiveness in computer vision [8]–[11], and natural language processing [12], [13]. In speech processing, SNNs have achieved promising performance in keyword spotting (KWS) [14]–[16], acoustic event and sound classification [17], [18], and automatic speech recognition (ASR) [19]–[22].

Despite promising results, SNNs are not without issues. For example, the challenges in training strategy [23], [24] and architectures [8], [25] prevent them from scaling up to large networks for complex real-world tasks, such as large-vocabulary ASR [19]. There have been many attempts to address the issues. By integrating self-attention mechanisms into SNN architecture [9], [11], [26], spiking Transformer models are proposed to enhances model's sequential modeling capability for long-range and complex temporal dependencies. The spiking Transformer employs multi-timestep firing to increase SNN's representational power by introducing a performance-efficiency tradeoff.

While several methods [27]–[30] are proposed to reduce the information loss in spike firing [31] and yields substantial performance gains with increase time windows, it also introduces computational complexity that scales linearly with window length, increasing both time and memory requirements [32]. This growing computational overhead ultimately limits the scope of large-scale SNN applications [33], [34]. Moreover, current research on spiking Transformers has predominantly focused on computer vision tasks [9], [11], [26], with their potential for speech processing remaining largely unexplored.

In this work, we aim to develop a spiking Transformer architecture for speech processing tasks with efficient training schemes. To mitigate the computational overhead of multi-timestep firing during training, we propose an *Input-aware Multi-Level Spike (IMLS)* firing mechanism. IMLS enables the simulation of multi-timestep dynamics within a single timestep, thereby reducing training costs while retaining the

Zeyang Song and Haizhou Li are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077

Haizhou Li is also with the School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen, 518172 China; Shenzhen Loop Area Institute, Shenzhen, China.

Shimin Zhang, Yuhong Chou and Jibin Wu are with the Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong SAR. Jibin Wu is also with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR([†]Corresponding Author: jibin.wu@polyu.edu.hk; haizhouli@cuhk.edu.cn)

spike-driven computational paradigm at inference. Moreover, IMLS introduces an input-aware adaptation strategy that adjusts neuronal firing thresholds according to the statistical distribution of pre-synaptic inputs, which improves both representational expressivity and training stability.

Building on IMLS, we design the *IML-Spikeformer*, a spiking Transformer tailored for speech processing. Its central component is the *Hierarchical Decay Re-parameterized Spiking Self-Attention (HD-RepSSA)* module, which enhances the representational capacity of the spiking attention map via a re-parameterization scheme while preserving spike-driven efficiency. Inspired by the auditory system's multi-scale temporal dynamics, we further propose a *Hierarchical Decay Mask (HDM)* that modulates attention maps across network layers, enabling the model to capture multi-scale temporal dependencies crucial for speech tasks.

Our main contributions are summarized as follows:

- **IMLS mechanism:** We introduce a multi-level spike firing strategy that reduces the training cost of multi-timestep firing, while an input-aware adaptive threshold aligned with pre-synaptic input statistics enhances model expressivity and stabilizes training.
- **IML-Spikeformer architecture:** We propose a novel spiking Transformer featuring the HD-RepSSA module for re-parameterized spiking self-attention and the HDM for hierarchical temporal modulation, enabling effective modeling of speech signals with multi-scale dynamics.
- **Scalability and efficiency:** We demonstrate that IML-Spikeformer scales to large speech processing tasks, achieving performance comparable to ANN-based Transformers while substantially reducing energy consumption across model sizes, highlighting its promise for energy-efficient speech processing.

The rest of the paper is organized as follows. Section II reviews the related works. Section III provides the preliminaries on spiking neuron model and spiking Transformer. Section IV and Section V present our core contributions: the IMLS firing mechanism and the HD-RepSSA module, respectively. Section VI details our experimental design and settings. Section VII presents experimental results demonstrating the effectiveness and efficiency of our IML-Spikeformer across ASR, speaker identification, and speaker verification tasks. Finally, Section VIII summarizes this work.

II. RELATED WORKS

We start by reviewing the existing SNN models for speech processing, which set the stage for the proposed IML-Spikeformer architecture.

A. Spiking Neural Networks for Speech Processing

SNNs offer the promises of energy efficiency, rapid response, and inherent robustness to noise perturbations. Motivated by these, various SNN architectures have been proposed, such as Spiking Multilayer Perceptrons (SMLP) [20], [35], Spiking Recurrent Neural Networks (SRNN) [36], [37], and Spiking Convolutional Neural Networks (SCNN) [15], [38]. These SNN architectures have demonstrated performance

comparable to that of conventional ANNs across various speech processing tasks, including KWS [39], [40], acoustic event and sound classification [17], [41], ASR [19]–[21], sound source localization [42], [43], speaker identification [16], and speech enhancement [44]–[46].

Despite these advancements, a significant performance gap persists between SNN models and state-of-the-art ANN models in large-scale speech processing tasks. This performance gap primarily stems from previous SNN architectures' difficulty scaling to larger networks [8], constraining their ability to process complex temporal dependencies and large data scale. Recently, spiking Transformer architecture was a response to the research problem. It shown promising performance on challenging computer vision tasks [9], [11], [22]. However, our preliminary studies indicate that existing spiking Transformer architectures experience significant performance degradation when applied directly to speech processing tasks where there is a need to process speech signals of variable lengths. This is primarily due to the inability of batch normalization in this condition. We consider IML-Spikeformer as an alternative in this paper.

B. Training Strategy for Large-Scale SNNs

In general, there are two primary approaches for training large-scale SNNs: Spatio-Temporal Backpropagation (STBP) [9], [26], [47] and ANN-to-SNN conversion (ANN2SNN) [22], [48], [49].

STBP explicitly unrolls the network dynamics of SNNs across T timesteps and L layers, applying backpropagation through time (BPTT) [23]. This multi-timestep simulation incurs $O(L \times T)$ memory complexity to store the neuronal states necessary for gradient computation, as well as $O(T)$ time complexity due to its inherently iterative processing. On the other hand, ANN2SNN methods convert a pre-trained ANN into SNN via rate-coding approximation, effectively circumventing the training overhead since the ANN2SNN conversion only need finetuning for few epochs. However, such technique necessitates many timesteps (like 16 [48], [49] to accurately approximate the activations of the ANN with spike firing rate, resulting in high computational costs during inference. Both methods introduce substantial computational overhead—STBP during training and ANN2SNN during inference—making the implementation of large-scale SNNs more expensive.

To resolve these limitations, we propose the IMLS firing mechanism, which effectively simulates multi-timestep neuronal firing within a single training timestep, thereby substantially enhancing the training efficiency of large-scale SNN. Also, our IMLS firing mechanism only need fewer timesteps(like 4 or 6) comparing to the ANN2SNN methods, enabling energy efficient and low-latency inference.

III. PRELIMINARIES

We next introduce the basic concepts of spiking neuron model and spiking Transformer, which are essential for understanding the proposed IML-Spikeformer architecture subsequently.

A. Spiking Neuron Models

Spiking neuron models are computational abstractions motivated by the understanding of biological neurons, acting as the basic computational units of SNNs. Among these, the Leaky Integrate-and-Fire (LIF) model [2] is widely adopted for constructing large-scale SNNs due to its mathematical tractability. The neuronal dynamics of the LIF model can be described by the following discrete-time formulation:

$$v^l[t] = \beta v^l[t-1] + x^{l-1}[t] - \theta s^l[t-1], \qquad (1)$$

$$s^l[t] = \mathcal{F}(v^l[t]) = \begin{cases} 0, & v^l[t] < \theta \\ 1, & v^l[t] \geq \theta \end{cases} \qquad (2)$$

where t denotes the timestep, θ is the firing threshold, $x^{l-1}[t]$ is the pre-synaptic input from layer $l-1$, and $v^l[t]$ and $s^l[t]$ denote the membrane potential and output spike in layer l, respectively. Equations 1-2 describe three fundamental processes of the spiking neuron: leakage & integration, reset, and firing.

Leakage & Integration: This process defines two essential dynamics within a spiking neuron: the decay of information according to a leaky factor $0 \leq \beta \leq 1$ and the integration of the pre-synaptic input $x^{l-1}[t]$. When $\beta = 1$, the neuron functions as an Integrate-and-Fire (IF) neuron with no information leakage between timesteps; otherwise, it operates as a LIF neuron.

Reset: After integration, the membrane potential undergoes soft reset by subtracting θ from neurons that fired in the previous timestep, as represented by the last term in Eq. 1.

Firing: The function $\mathcal{F}(\cdot)$ represents the spike firing mechanism. When the membrane potential $v^l[t]$ surpasses the firing threshold θ, an output spike $s^l[t] = 1$ is generated; otherwise, $s^l[t] = 0$.

B. Spiking Transformers

The spiking Transformer adapts the conventional Transformer architecture to enhance for computational efficiency. The family of spike-driven transformers is an example [11], [26], [47]. It typically comprises two key modules: Spike-driven Self-Attention (SDSA) and a Spiking Channel Multi-Layer Perceptron (ChannelMLP) [26]. An input sequence $X = \{x[1], x[2], \ldots, x[T]\}$ over whole time window is processed by these two modules,

$$\begin{aligned} X' &= \text{SDSA}(X) + X, \\ X'' &= \text{ChannelMLP}(X') + X'. \end{aligned} \qquad (3)$$

Specifically, the SDSA module concatenates a self-attention mechanism with a spiking neuron layer, denoted as $\mathcal{SN}(\cdot)$. This spiking neuron layer converts floating-point inputs into binary spikes. Taking SDSA-3 [11] (shown in Fig.1 (c) left) as an example:

$$\begin{aligned} \mathbf{Q} &= XW_Q, \quad \mathbf{K} = XW_K, \quad \mathbf{V} = XW_V, \\ \text{SDSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \mathcal{SN}(\mathbf{Q_s}\mathbf{K_s}^T\mathbf{V_s})W_{\text{out}}, \end{aligned} \qquad (4)$$

where W_Q, W_K, W_V represent the query, key, value transformation matrices. The query is defined as $\mathbf{Q_s} = \mathcal{SN}(\text{BN}(\mathbf{Q}))$,

with $\mathbf{K_s}$ and $\mathbf{V_s}$ defined analogously. The SDSA-3 implements linear attention through the direct multiplication of $\mathbf{Q_s}\mathbf{K_s}^T\mathbf{V_s}$, which can be computed as $\mathbf{Q_s}(\mathbf{K_s}^T\mathbf{V_s})$, achieving linear computational complexity with respect to sequence length and thereby enabling efficient long sequence modeling. $\text{BN}(\cdot)$ denotes the Batch Normalization layer, which is applied to normalize pre-synaptic inputs, facilitating stable information transmission. The results of QKV multiplication is then projected with a linear layer with weight W_{out} for the output of SDSA-3 module. Just like the MLP module in ANN transformers, the ChannelMLP module, a two-layer spiking MLP, is applied to spiking Transformer to capture the complex information across channels, formulated as:

$$\text{ChannelMLP}(X) = \mathcal{SN}(\mathcal{SN}(X)W_1)W_2. \qquad (5)$$

where W_1, W_2 are two learnable weights of the spiking MLP.

IV. INPUT-AWARE MULTI-LEVEL SPIKE FIRING (IMLS)

The spiking Transformer architecture that adopts multi-timestep firing mechanism, e.g. STBP, introduces significant computational overhead. Furthermore, such architecture typically employs a fixed firing threshold for spiking neurons, constraining their ability to represent the dynamically varying range of pre-synaptic inputs. To address these issues, we propose an Input-aware Multi-Level spike (IMLS) firing mechanism to enhance both computational efficiency and representation power.

A. Multi-Level Spike Firing

To mitigate the training overhead of the multi-timestep firing, we introduce the MLS firing mechanism that performs multi-timestep firing in a single timestep. For an IF neuron with soft reset, the equivalence between its T-timestep iterative firing and the proposed MLS firing is established in Fig. 1(a). Specifically, when receiving a pre-synaptic input only at the first timestep ($X = \{x[1]\}$), the IF neuron continues to emit spikes in subsequent timesteps until its membrane potential $v[t]$ falls below the firing threshold θ with soft reset, resulting in a spike train $\{s[t]\}, t \in [1, \ldots, T]$ (Fig. 1(a) left show the multi-timestep firing of the IF neuron with $T = 4$).

In contrast, the MLS firing mechanism condenses the multi-timestep iterative process into a single timestep operation. It represents the entire spike train from the IF neuron $\{s[t]\}$ as a single multi-level spike $s^M = \sum_{t=1}^{T} s[t]$ (Fig. 1(a) right), where the magnitude directly corresponds to the total number of spikes that would have been fired through the complete multi-timestep firing process. The multi-level spike can be formulated as:

$$s^M = \begin{cases} 0, & v[1] < \theta \\ n, & (n-1)\theta \leq v[1] < n\theta \\ T, & (T-1)\theta \leq v[1] \end{cases} \qquad (6)$$

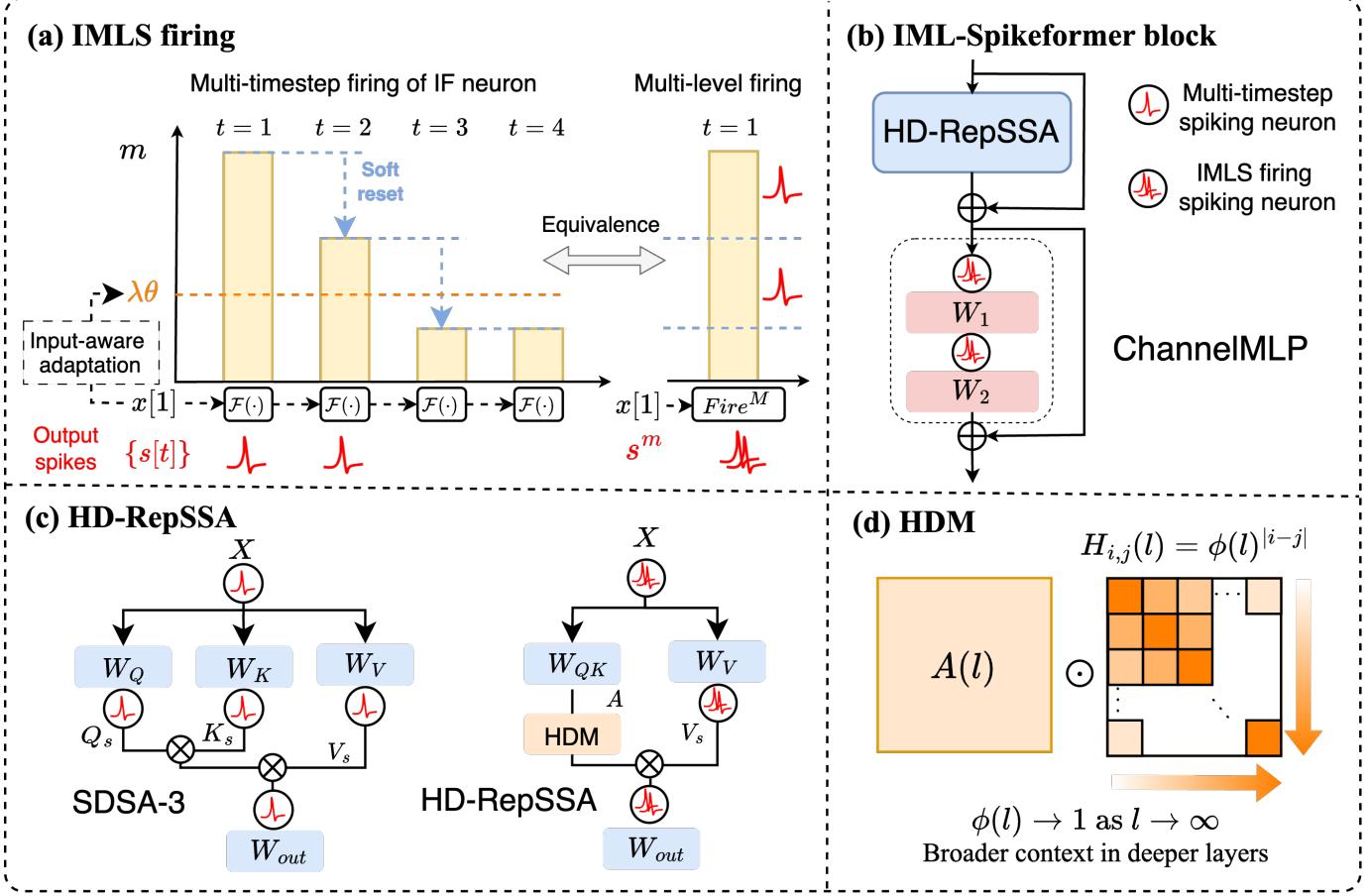$$= \mathcal{F}^M(v[1], \theta) = \lfloor \text{clip}(\frac{v[1]}{\theta}, 0, T) \rfloor,$$

Fig. 1. Overview of the proposed IML-Spikeformer architecture. (a) When only receive the pre-synaptic input in the first timestep $x[1]$, the membrane potential m iterative update of IF neuron is shown (left). The IMLS firing mechanism offers an equivalence between the output spike train $\{s[t]\}$ generated by the multi-timestep firing of an IF neuron and multi-level spike representation s^M (right) in a single timestep. The threshold $\lambda\theta$ dynamically adjust through the input-aware adaptive scaling factor λ, which modulates the base threshold θ in response to the pre-synaptic inputs $x[1]$. (b) An IML-Spikeformer block consists of two primary components: a HD-RepSSA module and a ChannelMLP module, where the IMLS firing spiking neurons are used for the spiking neuron layers $\mathcal{SN}(\cdot)$ in both modules. The IMLS firing spiking neuron use multi-level spike for efficient training, and converted to equivalent binary spike train for spike-driven inference. (c) Comparison between the SDSA-3 approach [11] with multi-timestep spiking neurons (left) and the proposed HD-RepSSA method (right). In the HD-RepSSA module, attention maps are calculated using re-parameterized weight W_{QK} and modulated by the HDM. (d) The attention map $A(l)$ in layer l is modulated by a HDM $H(l)$ governed by a layer-specific decay factor $\phi(l)$, which progressively increases (approaching 1) in deeper layers. The color intensity in the visualization corresponds to value magnitude, with deeper orange indicating values closer to 1.

where $n \in 1, \ldots, T-1$, $\lfloor \cdot \rfloor$ denotes the floor function, and $v[1]$ represents the membrane potential resulting from the integration of pre-synaptic input $x[1]$ at the initial timestep.

This multi-level spike representation in MLS shares similarities with burst coding [50]. Recent implementations [51]–[53] also employ spike magnitude to encode enhanced neural activity beyond binary spikes. However, the fundamental difference lies in their temporal dynamics: while burst coding only considers the number of spikes within a time interval, MLS establishes a mathematical equivalence between multi-timestep IF neuron firing and multi-level spike values. This equivalence (shown in Fig. 1(a)) enables seamless conversion from integer spike computing to binary spike-driven computing during inference.

Since the $\lfloor \cdot \rfloor$ is non-differentiable at the boundaries, we leverage the Generalized Straight-Through Estimator (G-

STE) [54] to approximate the gradient, yielding:

$$
\begin{aligned}
\frac{\partial s^M}{\partial v[1]} &= \mathbb{E}_{s^M}\left[\frac{\partial s^M}{\partial v[1]}\right] = \frac{\partial}{\partial v[1]}\mathbb{E}[s^M] \\
&= \frac{\partial \text{clip}(\frac{v[1]}{\theta}, 0, T)}{\partial v[1]} = \begin{cases} \frac{1}{\theta}, & \text{if } 0 \leq v[1] \leq \theta T \\ 0. & \text{Otherwise} \end{cases}
\end{aligned}
\tag{7}
$$

The proposed MLS establishes a memory-time tradeoff in training and inference: during GPU training, MLS utilizes multi-level spike s^M that compresses a spike train into a single timestep, enabling efficient training. During neuromorphic inference, MLS converts the multi-level spikes s^M into equivalent binary spike trains $\{s[t]\}$, preserving the spike-driven computation. This transformation maintains mathematical equivalence as follows,

$$
s^M W = [\sum_{t=1}^{T} s[t]]W = \sum_{t=1}^{T} s[t]W,
\tag{8}
$$

where dense matrix operations $s^M W$ during training become sparse accumulation $s[t]W$ during neuromorphic deployment. By strategically allocating computational resources, prioritizing temporal compression during training and spike-driven computations during inference, the MLS framework allows for memory and time saving during training while enabling efficient neuromorphic hardware deployment.

The MLS firing mechanism is also compatible with neuromorphic chips. In practice, an MLS neuron generating a multi-level spike of value k is implemented as emitting k consecutive binary spikes within a short time window, like burst coding [50] and graded spike firing [55]. Current neuromorphic chips like Intel's Loihi [4] and Speck [56] support this implementation strategy, making the MLS architecture naturally suited for energy-efficient neuromorphic deployment.

B. Input-aware Multi-Level Spike Firing

In the previous studies [9], [11], [26], the Batch Normalization(BN) layers are employed in spiking Transformer for vision tasks to normalize pre-synaptic inputs, ensuring the membrane potential remains within an appropriate range to sustain stable firing rates. However, BN becomes problematic for variable-length speech sequences because it computes statistics across the batch dimension, leading to inconsistent normalization when sequence lengths vary significantly [57]. This statistical inconsistency destabilizes training convergence. Conversely, removing BN entirely may result in unbounded pre-synaptic inputs that cause erratic firing patterns, especially in deeper SNNs.

For stable spike firing, we propose Input-aware Multi-Level Spike (IMLS) firing mechanism, which extends the MLS firing mechanism with adaptive thresholds that automatically adjust to pre-synaptic input, allowing our model to maintain stable spike firing patterns even without explicit normalization like BN. The IMLS firing mechanism incorporates a channel-wise adaptive scaling factor $\lambda \in \mathbb{R}^C$ for the firing threshold θ. Unlike fixed thresholds, the adaptive threshold $\lambda\theta$ dynamically adjusts in response to the statistical distribution of pre-synaptic inputs $x[1]$.

During training, for the i-th pre-synaptic input batch $x^i[1] \in \mathbb{R}^{B \times L \times C}$, where B, L, and C denote the batch size, sequence length, and number of channels respectively, λ is dynamically updated based on Λ_i. Here, Λ_i represents the maximum pre-synaptic inputs in each channel:

$$\Lambda_i = \max_{b\in[1,B],l\in[1,L]} x^i_{b,l,:}[1], \ \lambda = \frac{T}{\Lambda_i}, \qquad (9)$$

where T represents the time window and the division is performed element-wise.

However, computing the maximum operation for each inference imposes additional computational overhead, especially in neuromorphic devices. To mitigate this, we use a fixed scaling factor $\tilde{\lambda}$ with a running average $\tilde{\Lambda}$ during training with a momentum parameter α:

$$\tilde{\Lambda} = (1-\alpha) \cdot \tilde{\Lambda} + \alpha \cdot \Lambda_i, \ \tilde{\lambda} = \frac{T}{\tilde{\Lambda}}. \qquad (10)$$

During inference, this scaling factor $\tilde{\Lambda}$ only updated in training and fixed in inference is utilized to modulate the threshold, thereby eliminating the computational overhead while preserving the adaptive properties.

In summary, the input-aware threshold adaptation mechanism extends the dynamic range of spiking neurons by dynamically adjusting thresholds relative to input intensities, enabling informative spike responses across diverse signal distribution. This adaptability stabilizes firing rates despite varying input distributions, preventing both neuronal saturation and silence that commonly occur with fixed thresholds, serving as an implicit normalization mechanism. Our IMLS firing mechanism simultaneously achieves enhanced representational capacity through stable, normalized neural firing and improved computational efficiency via single-timestep processing, establishing IMLS as a robust foundation for training spiking Transformer architectures in large-scale speech processing tasks.

V. HIERARCHICAL DECAY RE-PARAMETRIZED SPIKING SELF-ATTENTION

As illustrated in Fig. 1 (b), each IML-Spikeformer block is composed of two key components: the HD-RepSSA module for token mixing and the Spiking ChannelMLP for channel mixing. At the core of IML-Spikeformer is the HD-RepSSA (shown in Fig. 1 (c)) — a novel spiking self-attention mechanism that precisely captures the hierarchical temporal dependencies inherent in speech signals while preserving the energy-efficient, spike-driven computation paradigm. The spiking neuron layers in HD-RepSSA $\mathcal{SN}(\cdot)$ utilize the IMLS firing spiking neurons introduced in the previous section, which use multi-level spike in training and binary spike train for spike driven inference.

A. Re-parametrized Spiking Self-Attention

In the vanilla self-attention mechanism from Transformer [58], the attention map is computed as the matrix product of the query and key: $\mathbf{A} = \mathbf{Q}\mathbf{K}^T$. This operation effectively measures token similarity, particularly when $\mathbf{Q}$ and $\mathbf{K}$ are continuous-valued. However, in spiking self-attention modules like SDSA-3, $\mathbf{Q}$ and $\mathbf{K}$ are converted to spike matrices $\mathbf{Q}_s$ and $\mathbf{K}_s$ via $\mathcal{SN}(\cdot)$. This conversion brings a significant limitation: the resulting attention map $\mathbf{A}_s = \mathbf{Q}_s\mathbf{K}_s^T$ struggles to accurately capture temporal relationships between tokens precisely.

To address this limitation, we introduce a novel RepSSA mechanism, which utilizes continuous-valued matrices for attention map calculation during training, while employing a re-parametrization technique to ensure spike-driven computations are preserved during inference:

$$\mathbf{A} = \begin{cases} XW_QW_K^TX^T, & \text{Training} \\ XW_{QK}X^T, & \text{Inference} \end{cases} \qquad (11)$$

$$\mathbf{V_s} = \mathcal{SN}(XW_V),$$

where X denotes the binary spike input. In the inference stage, we fuse the query and key transformation weights W_Q, W_K
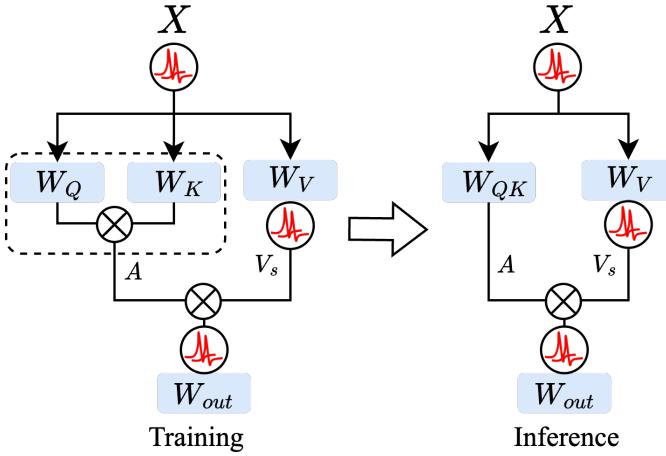
Fig. 2. Diagram illustrating the RepSSA module. During inference, the weight matrices W_Q and W_K are re-parametrized into a single matrix W_{QK} to maintain spike-driven computation.

as $W_{QK} = W_Q W_K^T$. Grounded on this re-parametrization, we define two versions of RepSSA as:

$$\begin{aligned} \text{RepSSA}_L(\mathbf{A}, \mathbf{V_s}) &= \mathcal{SN}(\mathbf{A}\mathbf{V_s})W_{out}, \\ \text{RepSSA}_S(\mathbf{A}, \mathbf{V_s}) &= \mathcal{SN}(\text{Softmax}(\frac{\mathbf{A}}{\sqrt{d_k}})\mathbf{V_s})W_{out}, \end{aligned} \quad (12)$$

where RepSSA_L and RepSSA_S respectively denote linear and softmax spiking self-attentions. d_k denotes the dimension of $\mathbf{K}$.

Since the token sequence length N in our experiments remains consistently smaller than the feature dimension D, linear attention, with a linear computational complexity of $O(ND^2)$ of sequence length N, offers no theoretical advantage over softmax attention, which has a complexity of $O(N^2D)$. Moreover, the linear attention mechanism introduces notable challenges in large-scale spiking Transformers, including performance degradation — phenomena primarily attributed to unbounded gradient issues [59]. Given these theoretical and practical considerations, we primarily adopt RepSSA_S in the proposed IML-Spikeformer. Nevertheless, we also evaluate RepSSA_L to explore the potential of linear spiking self-attention for speech processing tasks.

B. Hierarchical Decay Mask

Speech signals inherently exhibit multi-scale temporal structures, organized hierarchically from phonemes to complete utterances. Neurophysiological studies have demonstrated that auditory cortical processing employs progressively longer temporal integration windows as signals propagate through the cortical hierarchy [60], [61]. Drawing inspiration from this biological principle, we design a hierarchical decay mask (HDM) $\mathbf{H}$ to model multi-scale temporal dependencies by modulating the attenuation rate between tokens across different layers (shown in Fig.1 (d)). For two tokens at positions i and j in layer l, the corresponding attenuation rate is defined as:

$$\mathbf{H}_{i,j}(l) = \phi(l)^{|i-j|}, \quad (13)$$

where $\phi(l)$ represents the layer-specific decay function, formulated as $\phi(l) = 1 - 2^{-5-l}$. The term $|i-j|$ denotes the absolute positional distance between the tokens. The decay factor increases as the layer depth increases, enforcing a structured attention modulation strategy: shallow layers apply more pronounced attenuation to prioritize local interactions, while deeper layers retain decay values near 1, preserving long-range dependencies crucial for capturing global context.

The HDM can be integrated into the proposed RepSSA through element-wise multiplication with the attention map, creating a biologically-inspired attention mechanism that inherently captures multi-scale temporal dependencies. This integration forms the core component of IML-Spikeformer, which is termed HD-RepSSA, formulated as:

$$\text{HD-RepSSA}(\mathbf{A}, \mathbf{V_s}) = \text{RepSSA}(\mathbf{A'}, \mathbf{V_s}), \ \mathbf{A'} = \mathbf{A} \odot \mathbf{H}. \quad (14)$$

where $\odot$ is hadamard product.

In general, HDM models the multi-scale temporal dependencies inherent in speech processing, simultaneously capturing fine-grained local acoustic features while representing long-term temporal structures. This dual-scale capability enables IML-Spikeformer to dynamically adapt its processing across multiple temporal resolutions, effectively addressing the hierarchical organization that characterizes natural speech signals. The integration in Eq. (14) is applied to two RepSSA variants, resulting in HD-RepSSA$_S$ and HD-RepSSA$_L$, respectively.

VI. EXPERIMENTAL SETUP

We now provide an overview of the experimental setup. We first outline the configurations for three speech processing tasks: automatic speech recognition (ASR), speaker identification, and speaker verification. We then introduce the baseline models for each task. Finally, we describe how we quantitatively estimate the energy consumption during model inference.

A. Task Configurations

To assess the effectiveness and efficiency of the proposed IML-Spikeformer model, we conducted extensive experiments on large-scale speech processing tasks where SNNs historically achieved inferior performance compared to ANNs. In all experiments, IML-Spikeformer is implemented as a direct substitute for ANN Transformer.

ASR: The proposed IML-Spikeformer was evaluated on two public ASR databases: AiShell-1 [62], consisting of 170 hours of Mandarin speech data, and LibriSpeech-960 [63], comprising approximately 960 hours of English audiobook data. The experimental pipelines follow the established ESPnet ASR recipes [64] for data preparation, model training, and evaluation protocols. We employ ANN Transformers, IML-Spikeformer, and other baseline models (detailed in Section VI-B) as encoders, with a fixed 6-layer Transformer decoder without language model across all experiments to ensure fair comparison.

For evaluation, we assess model performance using Character Error Rate (CER) on AiShell-1's development and test sets

(labeled as "dev" and "test" in Table II). For LibriSpeech-960, we evaluate using Word Error Rate (WER) on both clean and other subsets of the development and test sets. The clean subset contains high-quality recordings with minimal background noise and clear pronunciation, while the other subset includes more challenging audio with background noise, varied recording conditions, and diverse speaker accents, providing comprehensive assessment across different acoustic conditions.

In data preprocessing, 80-dimensional log Mel filterbank features represent the acoustic inputs. Data augmentation techniques are systematically applied for improved robustness: SpecAugment [65] for frequency and time masking, and speed perturbation for temporal variability. The input features are subsequently downsampled through two consecutive 3×3 convolutional layers with stride 2, reducing the temporal dimensionality of the original speech inputs.

Speaker Identification and Verification: Experiments were also conducted on the VoxCeleb dataset, which includes Vox-Celeb1 [66] and VoxCeleb2 [67], containing over 2,000 hours of speech from interview videos on YouTube. Specifically, VoxCeleb1 consists of more than 100,000 utterances from 1,251 speakers, while VoxCeleb2 features over 1,000,000 utterances from 6,112 speakers. For speaker identification, we follow the official partition of VoxCeleb1, where the training set includes 138,361 utterances, and the test set contains 8,251 utterances from the 1,251 speakers. In the speaker verification experiments, the models were trained independently on VoxCeleb1 and VoxCeleb2 development sets, with evaluation performed on the VoxCeleb1-O test partition.

In all speaker identification and verification experiments, we implemented the pipeline with Transformer-based embedding extractors. All three approaches, namely ANN Transformer, IML-Spikeformer, and spike-driven Transformer baseline, maintain the same architectural configurations and training protocols as outlined in [68].

B. Baseline Models

We compare the effectiveness and energy efficiency of IML-Spikeformer against a range of ANN and SNN baseline models for the aforementioned speech tasks. For ANN baselines, we adopt the vanilla Transformer architecture for all three tasks. Additionally, we include an RNN-based model in ASR, that has a VGG network for feature extraction followed by a 3-layer bidirectional LSTM encoder with 1024 hidden and output dimensions, referred to as VGG-BiLSTM.

Due to the limited availability of directly comparable SNN baselines in large-scale speech processing, we re-implement several state-of-the-art SNN models as comparative baselines. We include the Spike-driven Transformer with SDSA-3 [11], the latest spiking Transformer architecture, with batch normalization layers removed to ensure convergence. We also implement two 3-layer SMLPs with 1024 hidden and output dimensions as encoders for ASR, incorporating either vanilla LIF neuron models or Gated Spiking Units (GSU) [44]. Notably, GSU enhances sequential modeling through its built-in gating mechanism, demonstrating superior performance in speech enhancement tasks [69].

To provide more comprehensive comparisons on the AiShell-1 ASR task, we include additional sequential processing architectures: binary S4D [70] and Spiking Temporal Convolution Network (Spiking TCN) [71]. These models represent different paradigms for handling temporal dependencies—S4D through structured state-space modeling and Spiking TCN through dilated convolutions with spiking dynamics. For the AiShell-1 task, we implement 6-layer architectures with 680 and 356 hidden dimensions for S4D and Spiking TCN, respectively.

For speaker identification, we include Spiking-LEAF [16], the only existing SRNN-based method applied to this task. For fair comparison, we used the same architecture in the proposed IML-Spikeformer, the Spike-driven Transformer baseline, and the traditional ANN Transformer. All models have the same number of parameters.

C. Energy Consumption Estimation

To assess the energy efficiency advantages of SNNs over ANNs, we adopt the energy estimation methods from the Intel N-DNS challenge [69]. This operation-based energy evaluation methodology is also commonly used in recent neuromorphic benchmarking studies [69], [72]. This approach uses the number of effective operations as a proxy for practical energy consumption, enabling reliable assessment of neuromorphic power advantages without requiring deployment on neuromorphic hardware.

Our evaluation framework separately assesses the energy consumption of ANNs and SNNs due to their different computational paradigms. ANNs rely on dense floating-point matrix computations implemented through Multiply-Accumulate (MAC) operations, while SNNs exploit sparse, event-driven processing using Accumulate (AC) operations for synaptic integration and neuronal dynamics. We calculated the detailed energy consumption based on previous studies on 45nm CMOS technology [73], where one floating-point MAC operation consumes $E_{\text{MAC}} = 4.6$ pJ while one AC operation consumes $E_{\text{AC}} = 0.9$ pJ.

For conventional ANNs, energy consumption scales linearly with the total floating-point computational load:

$$E_{\text{ANN}} = E_{FLOPs} = E_{\text{MAC}} \sum_{l=1}^{L} \text{FLOPs}^l \quad (15)$$

where the energy consumption of ANNs comes entirely from Floating Point Operations (FLOPs), FLOPs^l represents the floating-point operations in layer l, and L denotes the total network depth.

For SNNs, energy consumption comprises two components reflecting Synaptic Operations (SynOPs) and Neuronal Operations (NeuOPs):

$$E_{\text{SNN}} = E_{\text{SynOPs}} + E_{\text{NeuOPs}} \quad (16)$$

$$E_{\text{SynOPs}} = E_{\text{AC}} \sum_{l=1}^{L} \text{SynOPs}^l \quad (17)$$

$$E_{\text{NeuOPs}} = 10 \times E_{\text{AC}} \sum_{l=1}^{L} \mathcal{N}^l \quad (18)$$

TABLE I
RESULTS ON AISHELL-1 ASR TASK SHOWING CHARACTER ERROR RATE (CER) ON DEVELOPMENT AND TEST SETS, ENERGY CONSUMPTION, AND ENERGY SAVING RATIO FOR THE PROPOSED IML-SPIKEFORMER COMPARED TO ANN AND SNN BASELINES. THE ENERGY SAVING RATIO IS RELATIVE TO ANN TRANSFORMER (DENOTED AS "×1").

Model	SNN	Parameters	Timestep	dev	test	Energy(mJ)	Energy Saving
VGG-BiLSTM* [74]	✗	93.26M	1	9.7	10.7	33.42	× 0.31
Binary S4D* [70]	✗	21.00M	1	10.3	11.9	1.73	× 6.01
	✗	30.35M	1	5.6	5.9	10.39	× 1
Transformer [64]	✗	46.11M	1	5.3	5.6	15.58	× 0.67
	✗	61.89M	1	5.1	5.6	20.78	× 0.5
LIF*	✔	22.96M	1	13.8	15.7	0.73	× 14.19
spiking TCN* [71]	✔	23.35M	1	11.4	13.7	1.62	× 6.41
GSU* [44]	✔	60.74M	1	10.9	12.6	3.20	× 3.25
Spike-driven Transformer* [26]	✔	30.35M	6	10.2	11.9	1.98	× 5.24
	✔	30.35M	4	5.8	6.2	1.74	× 5.96
IML-Spikeformer	✔	30.35M	6	5.5	6.0	2.24	× 4.64
	✔	46.11M	6	5.3	5.7	3.12	× 3.32
	✔	61.89M	6	5.2	5.7	4.03	× 2.58

* Our reproduced results based on publicly available codebases.

The synaptic operations are quantified as $\text{SynOPs}^l = T \sum_i R_i^l \cdot C_i^l$, where T represents the simulation time window, and R_i^l and C_i^l denote the firing rate and incoming synaptic connections of presynaptic neuron i in layer l, respectively. The detailed calculations of attention modules' synaptic operations are presented in Table I of the Supplementary Materials. The neuronal operations are quantified by $\mathcal{N}^l$ active neurons per layer, where each neuron operation consumes approximately 10 times the energy of a synaptic operation on the Loihi architecture [69].

VII. EXPERIMENTAL RESULTS

A. Main Results

In this section, we report the main results of the proposed IML-Spikeformer alongside baseline models for comparison across three tasks: ASR, speaker identification, and speaker verification. We will make our code publicly available after the review process.

1) Automatic Speech Recognition:: Table I presents a comprehensive comparison of our IML-Spikeformer against both ANN and SNN baselines on the AiShell-1 ASR task. Our proposed IML-Spikeformer (30.35M parameters, 6 timesteps) achieves a Character Error Rate (CER) of 5.9% on the test set, outperforming existing SNN approaches by substantial margins. Notably, it reduces the test CER by 9.8% and 6.7% compared to the LIF-based model (15.7% CER) and the GSU baseline [44] (12.6% CER), respectively. Furthermore, IML-Spikeformer surpasses the Spike-driven Transformer baseline, achieving a 6.0% absolute CER reduction (from 11.9% to 5.9%). Compared with the ANN Transformer baseline results reported in the ESPnet toolkit [64], the proposed IML-Spikeformer achieves comparable CER on the test set while delivering substantial computational efficiency benefits—remarkably, a 4.64× reduction in energy consumption.

For the ASR results on the LibriSpeech-960 dataset, shown in Table II, IML-Spikeformer achieves a Word Error Rate

(WER) of 3.1% on development set and 3.4% on the test set — a performance on par with ANN Transformer baselines. Additionally, IML-Spikeformer significantly outperforms SMLPs with the LIF neuron model [20], alongside the GSU model [45] and spike-driven Transformer baselines, demonstrating its superiority in large-vocabulary ASR tasks.

2) Speaker Identification and Verification: Beyond ASR, IML-Spikeformer is also evaluated on speaker identification and verification tasks, with results illustrated in Table III and Table IV, respectively.

As demonstrated in Table III, the proposed IML-Spikeformer exhibits robust performance across multiple parameter configurations while maintaining substantial computational efficiency. It is observed that IML-Spikeformer achieves classification accuracies of 67.43%, 71.83%, and 74.34% for model sizes of 0.6M, 1.18M, and 1.76M parameters, respectively. Notably, the largest configuration surpasses its ANN Transformer counterpart, obtaining an improvement of 1.24% accuracy. In comparison with the SNN baseline, IML-Spikeformer significantly surpasses the SRNN-based implementation presented in [16] (67.43% versus 30.45%), despite utilizing fewer parameters.

Table IV presents speaker verification performance on the VoxCeleb1-O evaluation set. For models trained on the VoxCeleb1 development set, our IML-Spikeformer with 8 timesteps achieves an Equal Error Rate (EER) of 4.47%, substantially outperforming the Spike-driven Transformer baseline (5.65%) while approaching the performance of the ANN Transformer (4.44%). When trained on the larger VoxCeleb2 development set, IML-Spikeformer achieves a competitive 2.70% EER compared to the ANN Transformer's 2.66% and significantly surpassing the Spike-driven Transformer baseline (4.53%). Notably, across all experimental configurations, IML-Spikeformer maintains comparable EER to ANN counterparts while consuming only 36.9% of the energy (2.53 versus 6.85mJ).

TABLE II
RESULTS ON LIBRISPEECH-960 ASR TASK SHOWING WORD ERROR RATE (WER) ON DEVELOPMENT AND TEST SETS, ENERGY CONSUMPTION, AND ENERGY SAVING RATIO FOR THE PROPOSED IML-SPIKEFORMER COMPARED TO ANN AND SNN BASELINES. THE "DEV" AND "TEST" REFER TO THE WER IN DEVELOPMENT AND TEST SETS.

Model	SNN	Parameters	Timestep	dev (%)		test (%)		Energy(mJ)	Energy Saving
				clean	other	clean	other		
VGG-BiLSTM* [74]	✗	202.4M	1	7.2	18.9	7.3	19.7	38.63	× 0.85
Transformer [64]	✗	99.36M	1	**2.8**	**7.6**	**3.2**	**8.0**	32.82	× 1
LIF [20]	✔	-	1	-	-	9.94	-	-	-
GSU* [44]	✔	185.1M	1	12.4	30.4	12.8	32.1	8.59	× 3.82
Spike-driven Transformer* [26]	✔	99.36M	4	10.4	25.3	10.5	25.7	4.61	× 7.12
	✔	99.36M	6	8.7	20.7	8.9	22.3	6.02	× 5.45
IML-Spikeformer	✔	99.36M	4	3.5	8.4	3.9	8.7	5.67	× 5.79
	✔	99.36M	6	<u>3.1</u>	<u>8.3</u>	<u>3.4</u>	**7.9**	7.60	× 4.32

* Our reproduced results based on publicly available codebases.
- These results are not publicly available.

TABLE III
RESULTS ON VOXCELEB1 SPEAKER IDENTIFICATION TASK SHOWING TEST ACCURACY ON THE VOXCELEB1 TEST SET AND ENERGY CONSUMPTION FOR THE PROPOSED IML-SPIKEFORMER COMPARED TO ANN TRANSFORMER AND THE PREVIOUS PUBLISHED SNN WORK [16].

Model	Params	SNN	Timestep	Acc.(%)	Energy(mJ)
Spiking-LEAF [16]	0.9M	✔	1	30.45	0.072
Transformer*	0.6M	✗	1	67.49	0.65
	1.18M	✗	1	71.31	1.03
	1.76M	✗	1	73.10	1.37
IML-Spikeformer	0.6M	✔	6	67.43	0.18
	1.18M	✔	6	71.83	0.24
	1.76M	✔	6	74.34	0.34

* Our reproduced results based on publicly available codebases.

TABLE IV
RESULTS ON SPEAKER VERIFICATION TASK SHOWING TEST ACCURACY ON VOXCELEB1-O TEST SET, ENERGY CONSUMPTION FOR THE PROPOSED IML-SPIKEFORMER, TRANSFORMER, AND SPIKE-DRIVEN TRANSFORMER BASELINE [11]. RESULTS FOR MODELS TRAINED ON VOXCELEB1 AND VOXCELEB2 DEVELOPMENT SET ARE PRESENTED.

Model	SNN	Timestep	EER(%)	Energy(mJ)
Training on VoxCeleb1 dev set				
Transformer [68]*	✗	1	4.44	6.85
Spike-driven Transformer* [11]	✔	8	5.65	1.55
IML-Spikeformer	✔	6	4.75	1.58
	✔	8	4.47	2.37
Training on VoxCeleb2 dev set				
Transformer [68]*	✗	1	2.66	6.85
Spike-driven Transformer* [11]	✔	8	4.53	1.62
IML-Spikeformer	✔	6	3.20	1.62
	✔	8	2.70	2.53

* Our reproduced results based on publicly available codebases.

3) Scalability of IML-Spikeformer: As demonstrated in Table I, we compared our IML-Spikeformer against ANN transformers across three model sizes (30.35M, 46.11M, and 61.89M parameters) on the AISHELL-1 ASR task. The results show that CER decreases consistently for both model types as parameter count increases, with IML-Spikeformer maintaining competitive performance at each scale. Similarly, Table III reveals that our model achieves comparable or superior performance compared to ANN transformers across all model scales for speaker identification. These consistent results across different tasks and model sizes demonstrate both the parameter scalability and effectiveness of our IML-Spikeformer architecture.

B. Ablation Studies

As shown in Table V, we conduct ablation studies on the Aishell-1 ASR task to evaluate the contribution of key components in IML-Spikeformer across three critical dimensions: **Firing methods**: Replacing IMLS firing mechanism with MLS results in substantial performance degradation, increasing CER by 2.4% on test sets. The performance decline is even more pronounced when switching to iterative multi-timestep firing, with CER increases of 3.2%. Notably, the negligible energy differences between IMLS, MLS, and multi-timestep firing indicate that IMLS maintains comparable spike firing rate while delivering superior performance through its input-aware threshold adaptation. These results demonstrate the significant performance advantages conferred by the IMLS firing mechanism.

HD-RepSSA components: We ablate the contributions of individual components within our HD-RepSSA, including RepSSA and HDM. Removing the HDM (HD-RepSSA$_S$ → RepSSA$_S$) leads to a modest CER increase of 0.6% on test set. More significantly, replacing HD-RepSSA$_S$ with standard SDSA increases CER by 1.5%. These results confirm that while both components contribute positively, the RepSSA module with its enhanced attention map capacity delivers more substantial performance benefits.
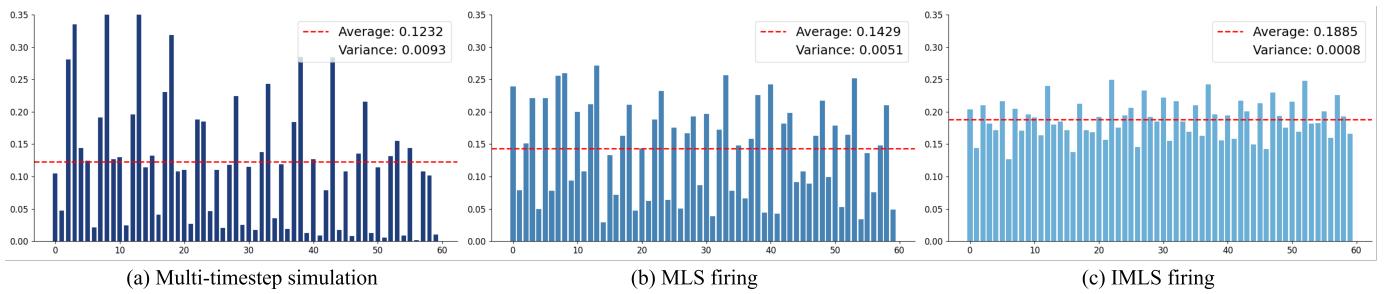
Fig. 3. Distribution of neuronal spike firing rates across network for three spike firing mechanisms on IML-Spikeformer. X-axis indicates the neuron indices. (a) Vanilla multi-timestep firing exhibits significant proportions of inactive neurons (near-zero firing rates), particularly in deeper layers. (b) IML-Spikeformer with MLS firing mechanism and fixed threshold demonstrates improved but still inconsistent activation patterns across network depth. (c) IML-Spikeformer with IMLS firing mechanism and input-aware threshold adaptation maintains stable firing rate distributions throughout all layers, with substantially reduced neuronal inactivity and more balanced activation patterns.

TABLE V
ABLATION STUDIES ON IML-SPIKEFORMER FOR THE AISHELL-1 ASR TASK. PERFORMANCE FOR VARIOUS MODEL CONFIGURATIONS ARE PRESENTED WITH RELATIVE CHANGES SHOWN IN BRACKETS.

Methods	Energy(mJ)	dev	test
IML-Spikeformer	2.24	5.5	6.0
Firing methods			
IMLS $\rightarrow$ MLS	2.20(-0.04)	7.6(+2.1)	8.3(+2.4)
IMLS $\rightarrow$ multi-timestep	2.16(-0.08)	8.2(+2.7)	9.1(+3.2)
HD-RepSSA components			
HD-RepSSA$_S$ $\rightarrow$ RepSSA$_S$	2.19(-0.5)	5.9(+0.4)	6.5(+0.6)
HD-RepSSA$_S$ $\rightarrow$ SDSA-3	1.99(-0.25)	8.1(+2.6)	9.0(+3.1)
Linear attention			
HD-RepSSA$_S$ $\rightarrow$ HD-RepSSA$_L$	2.06(-0.18)	6.1(+0.6)	6.7(+0.8)
HD-RepSSA$_S$ $\rightarrow$ RepSSA$_L$	2.01(-0.23)	7.9(+2.4)	8.7(+2.8)

Linear attention: We evaluated linear attention variants to analyze the accuracy-efficiency tradeoff. Switching from HD-RepSSA$_S$ to its linear counterpart (HD-RepSSA$_L$) increases CER by 0.8% while providing slight energy reduction, demonstrating the classic tradeoff between computational efficiency and model accuracy. The substantial performance gap between HD-RepSSA$_L$ and RepSSA$_L$ further demonstrates the critical role of HDM in maintaining speech processing performance, particularly under linear attention settings where performance degradation is more pronounced.

C. Spike Firing Rate Analysis

As evidenced in Table V, both MLS and IMLS yield performance improvements over multi-timestep firing, with IMLS demonstrating markedly superior results. To elucidate the underlying mechanism responsible for these performance improvements, we analyze neuronal spike firing rate distributions of each linear layer across network depths for three firing mechanisms on our IML-Spikeformer, as visualized in Fig. 3.

Fig. 3(a) reveals a critical limitation in the multi-timestep firing: a substantial proportion of spiking neurons exhibit near-zero firing rates across network layers, resulting in sparse activation patterns. This widespread neuronal inactivity substantially constrains the network's representational capacity, as established in previous study [31], directly accounting for

the model's performance degradation. Fig. 3(b) demonstrates that while MLS firing mechanism with the fixed threshold improves overall performance, it still exhibits considerable proportions of inactive neurons, indicating persistent representational constraints.

In contrast, the proposed IMLS firing mechanism directly addresses these limitations through input-aware threshold adaptation. By continuously monitoring pre-synaptic input distributions and dynamically calibrating neuronal firing thresholds to match these statistics, IMLS maintains appropriate firing rates throughout the network as shown in Fig. 3(c). This adaptive threshold mechanism effectively functions as an implicit normalization operation that eliminates the need for problematic BN layers. The resulting firing rate stabilization ensures consistent information propagation through all network depths, facilitating effective representation transformation across diverse speech patterns and enabling the robust performance scaling observed in our IML-Spikeformer architecture.

D. Attention Maps Analysis

The empirical results presented in Table V demonstrate a notable discrepancy in the performance impact of HDM across softmax and linear attention. When integrated with linear attention mechanisms, HDM yields a substantial CER reduction of 2.3% (RepSSA$_L$ $\rightarrow$ HD-RepSSA$_L$), whereas the corresponding improvement in softmax-based attention remains comparatively modest at 0.6% CER reduction (RepSSA$_S$ $\rightarrow$ HD-RepSSA$_S$). This differential impact suggests that HDM provides essential inductive bias that specifically addresses representational limitations inherent in linear attention formulations.

To elucidate the underlying mechanisms, Figure 4 presents a comparative visualization of attention maps across network depth. Figure 4(a) reveals that RepSSA$_L$ without HDM generates attention maps characterized by near-uniform weight distributions across all token pairs. This uniformity persists throughout the network hierarchy (layers 1, 6, and 12), indicating the model's inability to establish differentiated token relationships or develop context-sensitive representational focus, thereby limiting the model's capacity for contextual feature extraction.
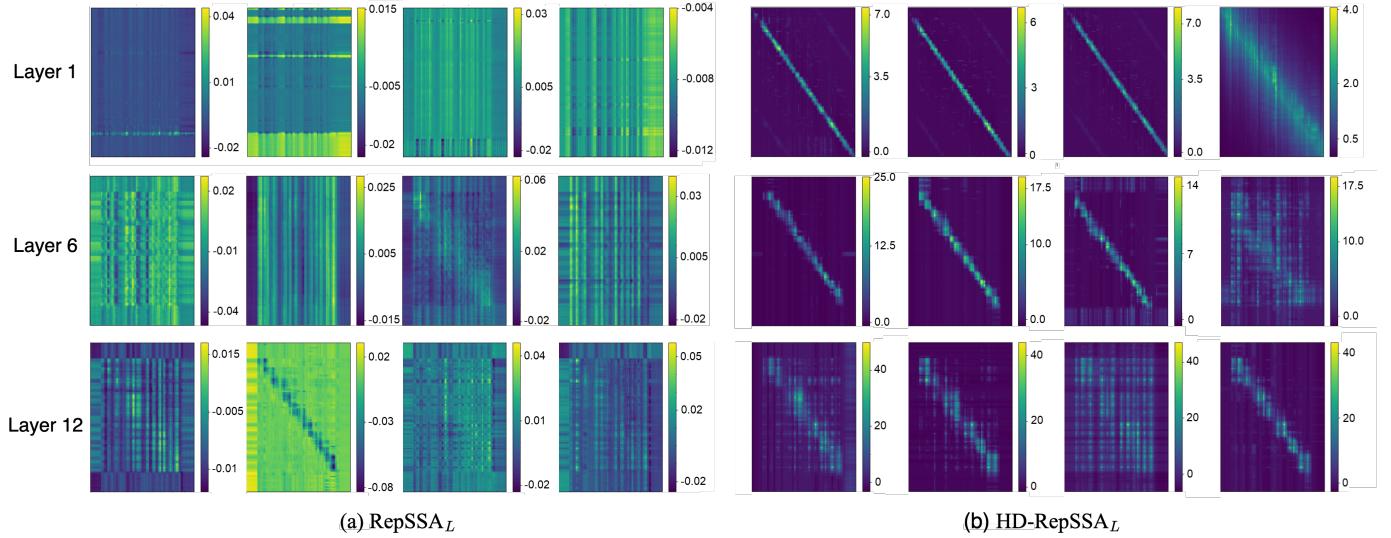
Fig. 4. Attention map visualization of across network depths in IML-Spikeformer with HD-RepSSA$_L$ and RepSSA$_L$. Attention maps of 4 heads shown at shallow (layer 1), intermediate (layer 6), and deep (layer 12) stages demonstrate how HDM enables hierarchical temporal dependencies, transitioning from local focus in shallow layers to global patterns in deeper layers.

In contrast, the HD-RepSSA$_L$ visualization in Figure 4(b) exhibits structured attention maps with layer-dependent characteristics. Early layers (e.g., layer 1) demonstrate concentrated diagonal activation, establishing localized temporal dependencies that correspond to phoneme-level processing. Intermediate and deeper layers (layers 6 and 12) gradually broaden their receptive fields, capturing hierarchically organized speech structures and acoustic features from phonemes to words to sentences. The expanding attention map range directly reflects the increasing receptive field size, where narrow attention spans in early layers capture phoneme-level acoustic details, medium-range attention in intermediate layers integrates phonemic sequences into word representations, and broad attention patterns in deeper layers enable sentence-level contextual understanding. This progressive expansion of the receptive field inherently supports the multi-scale temporal processing crucial for speech tasks, enabling the model to preserve local precision while capturing global context—both essential for effective speech representation. The hierarchical attention structure provides empirical explanation for the performance improvement observed in Table V, demonstrating how HDM utilizes structured inductive bias that compensates for the representational limitations of linear attention.

E. Training Efficiency Analysis

The IMLS firing mechanism not only stabilizes spike firing but also significantly enhances overall training efficiency and maintained fast convergence speed. Conventional multi-timestep iterative firing requires iterative firing across T timesteps, demanding storage of all intermediate states for BPTT, resulting in memory complexity of $O(L \times T)$ for an L-layer network. In contrast, IMLS computes multi-level spikes directly from the membrane potential at the initial timestep, eliminating extended timesteps and intermediate state storage, thereby reducing training and memory complexity to $O(L)$.
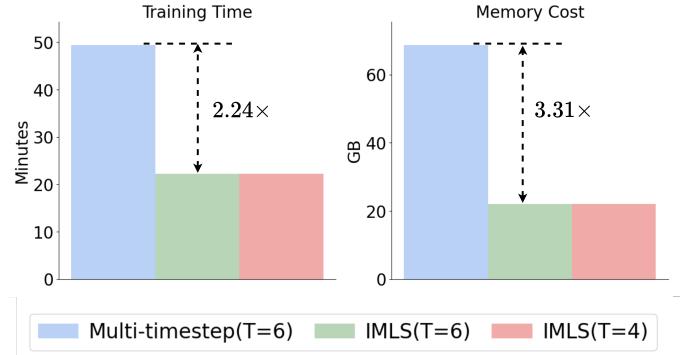

Fig. 5. The training time per epoch and the GPU memory cost of IML-Spikeformer with the iterative multi-timestep firing and our IMLS firing mechanisms, evaluated on 4 NVIDIA GeForce RTX 3090 Ti GPUs. The IMLS firing mechanism reduces computational overhead with single-timestep training.

As demonstrated in Fig. 5, when training IML-Spikeformer on 4 NVIDIA GeForce RTX 3090 Ti GPUs, IMLS achieves a $2.24\times$ reduction in per-epoch training time and cuts memory costs by $3.31\times$ compared to iterative multi-timestep firing. Notably, IMLS maintains constant training computational and memory costs regardless of the number of timesteps (T), as evidenced by comparing IMLS with T=4 and T=6. This characteristic makes IMLS especially valuable for our IML-Spikeformer architecture when addressing complex speech processing tasks that benefit from extended T to achieve optimal performance.

To establish the complete training efficiency picture, Fig. 1 in the Supplementary Materials shows that IMLS achieves comparable convergence speed to the multi-timestep baseline, ensuring that the computational optimizations do not come at the cost of learning effectiveness. Consequently, the combination of faster per-epoch training with equivalent convergence speed translates to significantly reduced overall training time

to convergence, establishing IMLS as a practically superior solution for efficient SNN training.

VIII. Conclusion

In this paper, we introduced IML-Spikeformer, a novel spiking Transformer architecture designed for large-scale speech processing. Our empirical evaluation confirms that IMLS enhances training efficiency while significantly improving performance and stability through adaptive threshold modulation. Through our proposed HD-RepSSA spiking self-attention module, IML-Spikeformer effectively overcomes the limited representational capacity of conventional spiking self-attention while successfully capturing the hierarchical temporal dependencies characteristic of speech signals. These innovations collectively establish IML-Spikeformer as a promising efficient framework for large-scale speech processing that achieves comparable performance to ANN transformers across ASR, speaker identification, and verification tasks while maintaining the efficiency benefits of SNNs.

References

[1] S. Latif, A. Zaidi, H. Cuayahuitl, F. Shamshad, M. Shoukat, and J. Qadir, "Transformers in speech processing: A survey," *arXiv preprint arXiv:2303.11607*, 2023.

[2] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[3] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[4] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[5] J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He *et al.*, "Towards artificial general intelligence with hybrid tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.

[6] P. Duan, Y. Ma, X. Zhou, X. Shi, Z. W. Wang, T. Huang, and B. Shi, "Neurozoom: Denoising and super resolving neuromorphic events and spikes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[7] C. Lenk, P. Hövel, K. Ved, S. Durstewitz, T. Meurer, T. Fritsch, A. Männchen, J. Küller, D. Beer, T. Ivanov *et al.*, "Neuromorphic acoustic sensing using an adaptive microelectromechanical cochlea with integrated feedback," *Nature Electronics*, vol. 6, no. 5, pp. 370–380, 2023.

[8] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep residual learning in spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 056–21 069, 2021.

[9] Z. Zhou, Y. Zhu, C. He, Y. Wang, S. Yan, Y. Tian, and L. Yuan, "Spikformer: When spiking neural network meets transformer," *arXiv preprint arXiv:2209.15425*, 2022.

[10] H. Zhang, Y. Li, B. He, X. Fan, Y. Wang, and Y. Zhang, "Direct training high-performance spiking neural networks for object recognition and detection," *Frontiers in Neuroscience*, vol. 17, p. 1229951, 2023.

[11] M. Yao, J. Hu, T. Hu, Y. Xu, Z. Zhou, Y. Tian, B. Xu, and G. Li, "Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips," *arXiv preprint arXiv:2404.03663*, 2024.

[12] R.-J. Zhu, Q. Zhao, G. Li, and J. K. Eshraghian, "Spikegpt: Generative pre-trained language model with spiking neural networks," *arXiv preprint arXiv:2302.13939*, 2023.

[13] X. Xing, Z. Zhang, Z. Ni, S. Xiao, Y. Ju, S. Fan, Y. Wang, J. Zhang, and G. Li, "Spikelm: Towards general spike-driven language modeling via elastic bi-spiking mechanisms," *arXiv preprint arXiv:2406.03287*, 2024.

[14] B. Cramer, Y. Stradmann, J. Schemmel, and F. Zenke, "Heidelberg spiking data sets for the systematic evaluation of spiking neural networks," in *Proceedings of the International Conference on Neuromorphic Systems*. Association for Computing Machinery, 2020, pp. 1–8.

[15] Q. Yang, Q. Liu, and H. Li, "Deep residual spiking neural network for keyword spotting in low-resource settings." in *Interspeech*, 2022, pp. 3023–3027.

[16] Z. Song, J. Wu, M. Zhang, M. Z. Shou, and H. Li, "Spiking-leaf: A learnable auditory front-end for spiking neural networks," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 226–230.

[17] J. Wu, Y. Chua, M. Zhang, H. Li, and K. C. Tan, "A spiking neural network framework for robust sound classification," *Frontiers in neuroscience*, vol. 12, p. 379777, 2018.

[18] S. Zhou, B. Yang, M. Yuan, R. Jiang, R. Yan, G. Pan, and H. Tang, "Enhancing snn-based spatio-temporal learning: A benchmark dataset and cross-modality attention model," *Neural Networks*, vol. 180, p. 106677, 2024.

[19] J. Wu, E. Yılmaz, M. Zhang, H. Li, and K. C. Tan, "Deep spiking neural networks for large vocabulary automatic speech recognition," *Frontiers in neuroscience*, vol. 14, p. 199, 2020.

[20] A. Bittar and P. N. Garner, "Surrogate gradient spiking neural networks as encoders for large vocabulary continuous speech recognition," *arXiv preprint arXiv:2212.01187*, 2022.

[21] W. Ponghiran and K. Roy, "Spiking neural networks with improved inherent recurrence dynamics for sequential learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 8001–8008.

[22] Q. Wang, T. Zhang, M. Han, Y. Wang, D. Zhang, and B. Xu, "Complex dynamic neurons improved spiking transformer network for efficient automatic speech recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 102–109.

[23] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018.

[24] Z. Wang, R. Jiang, S. Lian, R. Yan, and H. Tang, "Adaptive smoothing gradient learning for spiking neural networks," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 35 798–35 816.

[25] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, 2023.

[26] M. Yao, J. Hu, Z. Zhou, L. Yuan, Y. Tian, B. Xu, and G. Li, "Spike-driven transformer," *Advances in neural information processing systems*, vol. 36, pp. 64 043–64 058, 2023.

[27] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu, "Differentiable spike: Rethinking gradient-descent for training spiking neural networks," *Advances in neural information processing systems*, vol. 34, pp. 23 426–23 439, 2021.

[28] Q. Meng, M. Xiao, S. Yan, Y. Wang, Z. Lin, and Z.-Q. Luo, "Training high-performance low-latency spiking neural networks by differentiation on spike representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 444–12 453.

[29] S. Yang and B. Chen, "Snib: improving spike-based machine learning using nonlinear information bottleneck," *IEEE transactions on systems, man, and cybernetics: Systems*, vol. 53, no. 12, pp. 7852–7863, 2023.

[30] ——, "Effective surrogate gradient learning with high-order information bottleneck for spike-based machine intelligence," *IEEE transactions on neural networks and learning systems*, 2023.

[31] Y. Guo, Y. Chen, L. Zhang, X. Liu, Y. Wang, X. Huang, and Z. Ma, "Im-loss: information maximization loss for spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 156–166, 2022.

[32] W. Fang, Y. Chen, J. Ding, Z. Yu, T. Masquelier, D. Chen, L. Huang, H. Zhou, G. Li, and Y. Tian, "Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence," *Science Advances*, vol. 9, no. 40, p. eadi1480, 2023.

[33] M. Xiao, Q. Meng, Z. Zhang, D. He, and Z. Lin, "Online training through time for spiking neural networks," *Advances in neural information processing systems*, vol. 35, pp. 20 717–20 730, 2022.

[34] Q. Yang, J. Wu, M. Zhang, Y. Chua, X. Wang, and H. Li, "Training spiking neural networks with local tandem learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 662–12 676, 2022.

[35] Z. Song, Q. Liu, Q. Yang, Y. Peng, and H. Li, "Ed-skws: Early-decision spiking neural networks for rapid, and energy-efficient keyword spotting," *arXiv preprint arXiv:2406.12726*, 2024.

[36] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking

neurons," *Advances in neural information processing systems*, vol. 31, 2018.

[37] B. Yin, F. Corradi, and S. M. Bohté, "Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks," *Nature Machine Intelligence*, vol. 3, no. 10, pp. 905–913, 2021.

[38] E. Yilmaz and T. E.-G. Taha, "Deep spiking networks for auditory keyword spotting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 1717–1731, 2020.

[39] J. Wu, Y. Chua, and H. Li, "A biologically plausible speech recognition framework based on spiking neural networks," in *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[40] S. Zhang, Q. Yang, C. Ma, J. Wu, H. Li, and K. C. Tan, "Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, pp. 16 838–16 847, Mar. 2024.

[41] Z. Pan, Y. Chua, J. Wu, M. Zhang, H. Li, and E. Ambikairajah, "An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks," *Frontiers in neuroscience*, vol. 13, p. 493280, 2020.

[42] B.-Z. Li, S. H. Pun, M. I. Vai, A. Klug, and T. C. Lei, "Axonal conduction delay shapes the precision of the spatial hearing in a spiking neural network model of auditory brainstem," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 4238–4241.

[43] D. Zhang, S. Wang, A. Belatreche, W. Wei, Y. Xiao, H. Zheng, Z. Zhou, M. Zhang, and Y. Yang, "Spike-based neuromorphic model for sound source localization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 113 911–113 936, 2024.

[44] X. Hao, C. Ma, Q. Yang, J. Wu, and K. C. Tan, "Towards ultra-low-power neuromorphic speech enhancement with spiking-fullsubnet," *arXiv preprint arXiv:2410.04785*, 2024.

[45] X. Hao, C. Ma, Q. Yang, K. C. Tan, and J. Wu, "When audio denoising meets spiking neural network," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 1524–1527.

[46] Y. Du, X. Liu, and Y. Chua, "Spiking structured state space model for monaural speech enhancement," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 766–770.

[47] M. Yao, X. Qiu, T. Hu, J. Hu, Y. Chou, K. Tian, J. Liao, L. Leng, B. Xu, and G. Li, "Scaling spike-driven transformer with efficient spike firing approximation training," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[48] J. Wu, C. Xu, X. Han, D. Zhou, M. Zhang, H. Li, and K. C. Tan, "Progressive tandem learning for pattern recognition with deep spiking neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7824–7840, 2021.

[49] M. Bal and A. Sengupta, "Spikingbert: Distilling bert to train spiking language models using implicit differentiation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 10, 2024, pp. 10 998–11 006.

[50] E. M. Izhikevich, N. S. Desai, E. C. Walcott, and F. C. Hoppensteadt, "Bursts as a unit of neural information: selective communication via resonance," *Trends in neurosciences*, vol. 26, no. 3, pp. 161–167, 2003.

[51] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.

[52] Y. Li and Y. Zeng, "Efficient and accurate conversion of spiking neural network with burst spikes," *arXiv preprint arXiv:2204.13271*, 2022.

[53] X. Chen, Q. Yang, J. Wu, H. Li, and K. C. Tan, "A hybrid neural coding approach for pattern recognition with spiking neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 5, pp. 3064–3078, 2023.

[54] Z. Liu, K.-T. Cheng, D. Huang, E. P. Xing, and Z. Shen, "Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4942–4952.

[55] S. B. Shrestha, J. Timcheck, P. Frady, L. Campos-Macias, and M. Davies, "Efficient video and audio processing with loihi 2," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 13 481–13 485.

[56] M. Yao, O. Richter, G. Zhao, N. Qiao, Y. Xing, D. Wang, T. Hu, W. Fang, T. Demirci, M. De Marchi *et al.*, "Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip," *Nature Communications*, vol. 15, no. 1, p. 4464, 2024.

[57] J. Wang, J. Wu, and L. Huang, "Understanding the failure of batch normalization for transformers in nlp," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 617–37 630, 2022.

[58] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[59] Z. Qin, X. Han, W. Sun, D. Li, L. Kong, N. Barnes, and Y. Zhong, "The devil in linear transformer," *arXiv preprint arXiv:2210.10340*, 2022.

[60] G. Hickok and D. Poeppel, "The cortical organization of speech processing," *Nature reviews neuroscience*, vol. 8, no. 5, pp. 393–402, 2007.

[61] J. P. Rauschecker and S. K. Scott, "Maps and streams in the auditory cortex: nonhuman primates illuminate human speech processing," *Nature neuroscience*, vol. 12, no. 6, pp. 718–724, 2009.

[62] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[63] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[64] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[65] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[66] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[67] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[68] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H.-y. Lee, and H. Meng, "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," *arXiv preprint arXiv:2203.15249*, 2022.

[69] J. Timcheck, S. B. Shrestha, D. B. D. Rubin, A. Kupryjanow, G. Orchard, L. Pindor, T. Shea, and M. Davies, "The intel neuromorphic dns challenge," *Neuromorphic Computing and Engineering*, vol. 3, no. 3, p. 034005, 2023.

[70] M.-I. Stan and O. Rhodes, "Learning long sequences in spiking neural networks," *Scientific Reports*, vol. 14, no. 1, p. 21957, 2024.

[71] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[72] J. Yik, K. Van den Berghe, D. den Blanken, Y. Bouhadjar, M. Fabre, P. Hueber, W. Ke, M. A. Khoei, D. Kleyko, N. Pacik-Nelson *et al.*, "The neurobench framework for benchmarking neuromorphic computing algorithms and systems," *Nature communications*, vol. 16, no. 1, p. 1545, 2025.

[73] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[74] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

Supplementary Materials

IML-Spikeformer: Input-aware Multi-Level Spiking Transformer for Speech Processing

A. Computing Infrastructure

All experiments are conducted on Ubuntu 20.04.5 LTS server equipped with NVIDIA GeForce RTX 3090 GPUs (24G Memory), Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz, Pytorch 1.13.0, and CUDA 11.8.

B. Detailed energy consumption

In this section, we present the detailed energy calculation of attention modules. Table 1 show the E_{FLOPs} of the Vallina Self-Attention(VSA) and E_{SynOPs} of SDSA-3, HD-RepSSA$_S$ and HD-RepSSA$_L$, respectively.

TABLE VI

ENERGY CONSUMPTION OF SELF-ATTENTION MODULES. T, N, D ARE SIMULATION TIMESTEP, TOKEN NUMBER AND INPUT DIMENSION. $R_C, \hat{R}, R_A$ DENOTE THE AVERAGE SPIKE FIRING RATES OF VARIOUS SPIKE MATRICES.

	VSA	SDSA-3	HD-RepSSA$_S$	HD-RepSSA$_S$
Q, K, V	$3ND^2 \cdot E_{MAC}$	$T \cdot R_C \cdot 3ND^2 \cdot E_{AC}$	$T \cdot R_C \cdot ND^2 \cdot E_{AC}$	$T \cdot R_C \cdot ND^2 \cdot E_{AC}$
$f(Q, K, V)$	$2N^2D \cdot E_{MAC}$	$T \cdot \hat{R} \cdot ND^2 \cdot E_{AC}$	$T \cdot \hat{R} \cdot 2N^2D \cdot E_{AC}$	$T \cdot \hat{R} \cdot 2ND^2 \cdot E_{AC}$
HDM	-	-	$T \cdot R_A \cdot N^2 \cdot E_{MAC}$	$T \cdot R_A \cdot N^2 \cdot E_{MAC}$
Scale	$N^2 \cdot E_{MAC}$	-	-	-
Softmax	$2N^2 \cdot E_{MAC}$	-	$2N^2 \cdot E_{MAC}$	-
Linear	$OP_{\text{MLP}} \cdot E_{MAC}$	$T \cdot R_C \cdot OP_{\text{MLP}} \cdot E_{AC}$	$T \cdot R_C \cdot OP_{\text{MLP}} \cdot E_{AC}$	$T \cdot R_C \cdot OP_{\text{MLP}} \cdot E_{AC}$

In Table 1, the OP_{MLP} refers to the number of operations in the ChannelMLP, $OP_{\text{MLP}} = 2D \cdot d_h$ for MLP with input dimension of D and hidden dimension d_h.

C. Confidence interval of the speaker identification task

In this section, we evaluate the speaker identification task across 5 independent runs with different random seeds to ensure statistical reliability, as shown in Table VII. All results are reported as mean $\pm$ standard deviation. The Spiking-LEAF results are taken directly from the original literature. As demonstrated in Table VII, our IML-Spikeformer consistently achieves performance comparable to or better than the Transformer baseline while consuming significantly less energy, demonstrating the superior stability and efficiency of our IML-Spikeformer.

D. Training convergence

In this section we provide the convergence speed comparison between the our IML-spikeformer with IMLS and the multi-timestep firing in Fig. 6. This figure shows that our model with IMLS can achieve better performance with comparable converge speed to their performance limits.

TABLE VII

RESULTS ON VOXCELEB1 SPEAKER IDENTIFICATION TASK OF 5 RUNS WITH RANDOM SEEDS.

Model	Params	SNN	Timestep	Acc.(%)	Energy(mJ)
Spiking-LEAF	0.9M	✔	1	30.45	0.072
	0.6M	✗	1	67.21±0.33	0.65±0
Transformer*	1.18M	✗	1	70.80±0.42	1.03±0
	1.76M	✗	1	72.81±0.25	1.37±0
	0.6M	✔	6	67.11±0.38	0.16±0.05
IML-Spikeformer	1.18M	✔	6	71.53±0.28	0.22±0.06
	1.76M	✔	6	73.84±0.23	0.30±0.09

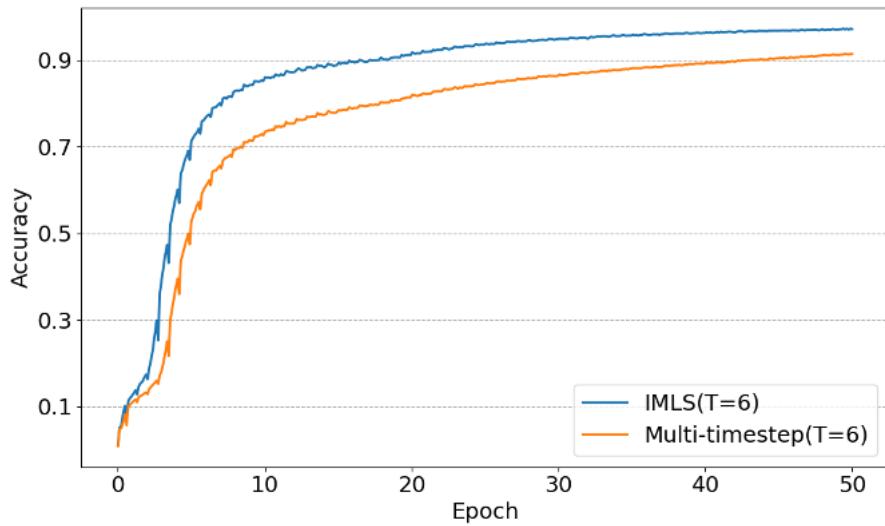* Our reproduced results based on publicly available codebases.

Fig. 6. Learning curves comparing IML-Spikeformer with iterative multi-timestep firing and our IMLS firing mechanisms. The curves demonstrate comparable convergence speed while IMLS achieves better final performance.