HetSyn: Versatile Timescale Integration in Spiking Neural Networks via Heterogeneous Synapses

Zhichao Deng[1,2,3]*, **Zhikun Liu**[1,2,3]*
Junxue Wang[1,2,3]*, **Shengqian Chen**[1,2,3], **Xiang Wei**[1,2,3], **Qiang Yu**[1,2]†
[1]School of Artificial Intelligence, CCA Lab, Tianjin University, Tianjin, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin, China
[3]School of Computer Science and Technology, Tianjin University, Tianjin, China
{dengzhichao, zk_liu813, wangjunxue, yuqiang}@tju.edu.cn

Abstract

Spiking Neural Networks (SNNs) offer a biologically plausible and energy-efficient framework for temporal information processing. However, existing studies overlook a fundamental property widely observed in biological neurons—synaptic heterogeneity, which plays a crucial role in temporal processing and cognitive capabilities. To bridge this gap, we introduce HetSyn, a generalized framework that models synaptic heterogeneity with synapse-specific time constants. This design shifts temporal integration from the membrane potential to the synaptic current, enabling versatile timescale integration and allowing the model to capture diverse synaptic dynamics. We implement HetSyn as HetSynLIF, an extended form of the leaky integrate-and-fire (LIF) model equipped with synapse-specific decay dynamics. By adjusting the parameter configuration, HetSynLIF can be specialized into vanilla LIF neurons, neurons with threshold adaptation, and neuron-level heterogeneous models. We demonstrate that HetSynLIF not only improves the performance of SNNs across a variety of tasks—including pattern generation, delayed match-to-sample, speech recognition, and visual recognition—but also exhibits strong robustness to noise, enhanced working memory performance, efficiency under limited neuron resources, and generalization across timescales. In addition, analysis of the learned synaptic time constants reveals trends consistent with empirical observations in biological synapses. These findings underscore the significance of synaptic heterogeneity in enabling efficient neural computation, offering new insights into brain-inspired temporal modeling. Code available at: https://github.com/dzcgood/HetSyn.

1 Introduction

Spiking Neural Networks (SNNs) have been widely studied as a promising and energy-efficient alternative to conventional artificial neural networks (ANNs), offering a biologically plausible computing paradigm characterized by sparse, event-driven signaling and inherent capability for temporal processing [1, 2, 3]. However, due to the complex dynamical characteristics of spiking neurons, effectively training SNNs and improving their performance remain major challenges in the field. This calls for new learning paradigms, either by adapting established methods from ANNs [4, 5, 6, 7, 8, 9, 10] or by drawing inspiration from biological mechanisms [11, 12, 13, 14, 15, 16].

While ANN-derived methods have led to notable progress, we instead focus on strengthening the biological foundations of SNNs by revisiting fundamental neurobiological mechanisms, which

*Equal contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

remain underexplored and hold great potential for enabling more versatile timescale integration in spiking systems. This capability is particularly important, as effective timescale integration is central to temporal cognition, and many real-world tasks—such as working memory, speech recognition, and sequential decision-making—require neural systems to operate across multiple timescales [17, 18, 19, 20]. Achieving such multi-timescale processing requires neural models to incorporate diverse temporal dynamics across different input pathways. To this end, various forms of temporal heterogeneity have been introduced in SNNs to enhance their capability to represent and integrate information across multiple time constants, such as ALIF [12], PLIF [21], neuron heterogeneity [22], and dendritic heterogeneity [23]. Although these approaches provide valuable insights, they remain limited to the neuron level or sub-neuronal level, where temporal dynamics are uniformly applied across aggregated inputs. This restricts their ability to assign distinct temporal responses to different synaptic pathways, limiting performance on tasks with complex temporal structure.

In contrast, numerous neuroscience studies have shown that synapses vary substantially across brain regions and cell types [24, 25], giving rise to a diverse temporal basis that supports multi-timescale integration and cognitive abilities [26, 27] (see Fig. 1A, B). This diversity, known as synaptic heterogeneity, constitutes a fundamental biological principle that plays a crucial role in shaping neural computation. Despite its biological significance, synaptic heterogeneity has rarely been incorporated into the design of spiking neural networks. Its computational potential remains largely unexplored, in part due to the challenges of modeling fine-grained, synapse-specific temporal dynamics.

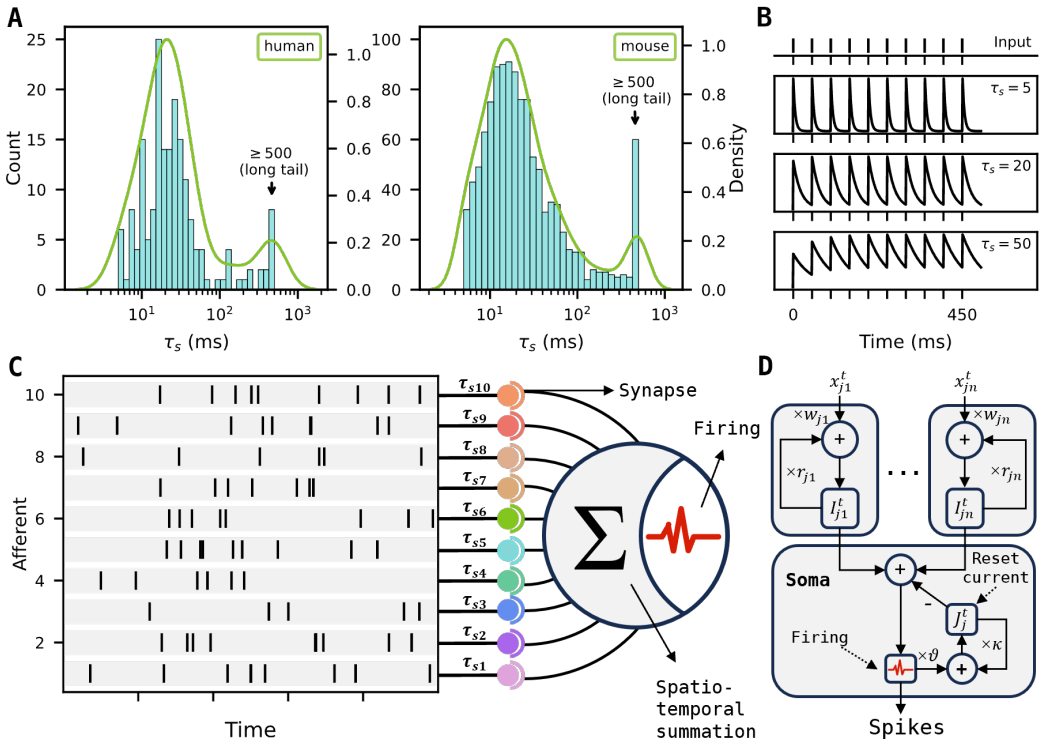

Figure 1: (A) Distributions of synaptic time constants (τ_s) extracted from human (left, 194 pairs) and mouse (right, 1213 pairs) cortex, shown as histograms with overlaid KDE curves (green). The broad, long-tailed ($\tau_s \geq 500$ ms) profiles reflect substantial synaptic heterogeneity. (B) Postsynaptic membrane potential traces under different synaptic time constants (τ_s = 5, 20, 50 ms, from second to fourth row) in response to a regular spike train (first row). (C-D) Schematic of HetSyn, in which afferent spikes are integrated via synapses with heterogeneous decay dynamics.

To bridge this gap, we take a first step toward incorporating synaptic heterogeneity into SNNs and introduce HetSyn by focusing on one key aspect: the diversity of synaptic time constants, motivated by our analysis of the raw data from the Allen Institute's recently released Synaptic Physiology

Dataset [28], which reveals that synaptic time constants follow a log-normal distribution across populations, providing direct empirical support for synapse-specific time constants (Fig. 1A). As illustrated in Fig. 1B, synapses with longer time constants are able to retain input-driven activity over extended periods, whereas those with smaller time constants respond more transiently and can effectively filter out noise or irrelevant fluctuations. This functional diversity lays the foundation for versatile timescale integration, enabling neural models to process information across both long and short temporal windows.

Furthermore, instead of relying on a single membrane time constant, HetSyn computes the membrane potential by aggregating synaptic currents, each governed by a synapse-specific decay factor. This synapse-driven formulation allows temporal integration to arise from the diverse dynamics of individual synapses, rather than being uniformly controlled at the neuron level (Fig. 1C, D). We instantiate HetSyn as HetSynLIF and demonstrate that it subsumes several representative spiking neuron models as special cases through parameter configuration (see Methods).

Our contributions are as follows:

- We propose HetSyn, the first modeling framework to explicitly explore synaptic heterogeneity in SNNs, offering a biologically plausible yet computationally powerful approach for versatile timescale integration.
- We demonstrate that HetSyn serves as a unified and extensible framework, capable of representing a wide range of existing spiking neuron models.
- We instantiate HetSyn as HetSynLIF and demonstrate its effectiveness across multiple temporal tasks, with consistently strong performance. Notably, we achieve 92.36% accuracy on the SHD dataset, which is, to the best of our knowledge, the best-reported accuracy among models with similar network structures.

2 Related Work

2.1 Training methods

There are two primary approaches for training SNNs: ANN-to-SNN conversion (ANN2SNN) [29, 30, 31, 32, 33, 34] and direct training using surrogate gradient methods [10, 35, 36, 37, 38, 39]. The ANN2SNN method first trains a conventional ANN and then maps its parameters to an SNN, where the firing rates of spiking neurons are used to approximate the continuous activations of the original ANN. However, it often suffers from accuracy degradation due to the approximation process. In comparison, surrogate gradient methods approximate the non-differentiable spiking function with a surrogate, thus enabling direct end-to-end training of SNNs via backpropagation through time (BPTT). This approach forms the basis of modern surrogate-gradient based training frameworks for SNNs, such as Spatio-Temporal Backpropagation (STBP) [35] and Slayer [36]. In this paper, we adopt the surrogate gradient method to enable efficient training of the HetSynLIF model.

2.2 Heterogeneity in SNNs

Recent studies have highlighted the significance of heterogeneity in SNNs, exploring its impact on network performance and temporal processing from various perspectives. For instance, neuron-level heterogeneity has been shown to enhance the stability and noise robustness of SNNs. Perez-Nieves et al. [22] show that networks with membrane time constants drawn from a Gamma distribution better capture complex temporal patterns, outperforming homogeneous networks in tasks with rich temporal structure. In addition, PLIF [21] learns a shared membrane time constant for each layer, which can be interpreted as introducing heterogeneity at a coarser granularity. Another source of heterogeneity arises from the spiking history of a neuron [12]. Firing thresholds that evolve over time based on recent spiking activity introduce a form of adaptive, history-dependent threshold heterogeneity. Furthermore, heterogeneity has also been explored at the level of neural dynamics and plasticity mechanisms. Chakraborty et al. [40] investigate the effects of heterogeneity in LIF and spike-timing-dependent plasticity (STDP) parameters. Additionally, the temporal heterogeneity of dendritic branches [23] has been integrated into SNNs. While considerable work has explored heterogeneity in SNNs, synapse-level heterogeneity remains largely underexplored. In contrast, our HetSyn introduces synapse-specific decay factors, embedding heterogeneity directly at the synaptic

level. This fine-grained design enables HetSyn to integrate temporal information over a broader range of timescales, thereby enhancing both computational flexibility and biological plausibility.

3 Methods

3.1 Vanilla LIF

Spiking neurons are the fundamental computational units in SNNs, enabling the modeling of temporal dynamics through biologically inspired mechanisms. Among various neuron models, the LIF neuron is one of the most widely adopted for its balance between computational efficiency and biological plausibility. It captures essential neuronal properties such as membrane potential leakage, input integration, and spike emission. The dynamics of the LIF neuron can be formulated as follows:

$$\frac{dV}{dt} = -\frac{V - V_{\text{rest}}}{\tau_{\text{m}}} + \sum_{i,j} w_i \cdot \delta\left(t - t_i^j\right) - \vartheta \cdot \sum_j \delta\left(t - t_{\text{s}}^j\right) \tag{1}$$

where V is the membrane potential, V_{rest} is the resting potential, and τ_{m} represents the membrane time constant. t_i^j and t_{s}^j denote the timing of the j-th input spike from the i-th presynaptic neuron and the j-th output spike from the postsynaptic neuron, respectively. The term w_i denotes the synaptic efficacy of the i-th afferent, ϑ denotes the firing threshold and $\delta(\cdot)$ represents the Dirac delta function. We set $V_{\text{rest}} = 0$ in this paper, and an equivalent continuous-time solution to Eq. (1) is then given by:

$$V^t = \sum_i w_i \sum_j k\left(t - t_i^j\right) - \vartheta \cdot \sum_j k\left(t - t_{\text{s}}^j\right) \tag{2}$$

where V^t is the membrane potential at time t. The term $k\left(t - t_i^j\right) = \exp(-\frac{t - t_i^j}{\tau_{\text{m}}})$, for $t > t_i^j$, is the synaptic kernel induced by the j-th spike of the i-th afferent. It describes the resulting postsynaptic potential (PSP), which decays exponentially at a rate governed by the membrane time constant τ_{m}. In practice, simulations typically adopt a discrete-time formulation, which is given by:

$$V^t = \rho \cdot V^{t-1} + \sum_i w_i \cdot z_i^t - \vartheta \cdot z^{t-1} \tag{3}$$

$$z^t = \text{H}(V^t - \vartheta) \tag{4}$$

where $\rho = \exp(-\frac{\Delta t}{\tau_{\text{m}}})$ denotes the membrane decay factor, with Δt representing the discrete timestep. The output spike z^t is computed using the Heaviside step function $\text{H}(\cdot)$.

3.2 LIF-based spiking neuron with HetSyn

We apply the HetSyn principle to the vanilla LIF neuron and propose HetSynLIF, a variant that incorporates synaptic heterogeneity. In HetSynLIF, the traditional membrane time constant is replaced by synapse-specific time constants. Instead of applying a uniform temporal filter to all inputs, each synaptic input is individually integrated through its own exponential decay, resulting in heterogeneous synaptic currents. This design shifts temporal integration from the membrane dynamics to the synaptic dynamics, reproducing synaptic diversity in biological systems. Moreover, we model the reset mechanism as a negative current injection. The neuron then accumulates these individually filtered synaptic currents and subtracts the reset current to update its membrane potential. The dynamics of HetSynLIF can be derived from Eq. (2), yielding:

$$V^t = \sum_i I_i^t - J^t \tag{5}$$

$$\frac{dI_i^t}{dt} = -\frac{I_i^t}{\tau_{\text{s},i}} + \sum_j w_i \cdot \delta\left(t - t_i^j\right) \tag{6}$$

$$\frac{dJ^t}{dt} = -\frac{J^t}{\tau_J} + \sum_j \vartheta \cdot \delta\left(t - t_{\text{s}}^j\right) \tag{7}$$

where I_i^t denotes the synaptic current from the i-th afferent at time t, governed by its own synapse-specific time constant $\tau_{\text{s},i}$, thereby capturing the heterogeneity in synaptic temporal dynamics. J^t

denotes the reset current, a neuron-level term that models the effect of spike-triggered reset and decays with time constant τ_J. The discrete-time formulations of synaptic and reset currents are given by:

$$I_i^t = r_i \cdot I_i^{t-1} + w_i \cdot z_i^t \tag{8}$$

$$J^t = \kappa \cdot J^{t-1} + \vartheta \cdot z^{t-1} \tag{9}$$

where $r_i = \exp(-\frac{\Delta t}{\tau_{s,i}})$ and $\kappa = \exp(-\frac{\Delta t}{\tau_J})$ are the decay factors for the i-th synaptic current and the reset current of the neuron, respectively.

3.3 Generalize to other spiking neurons

Equipped with HetSyn, our HetSynLIF model is highly flexible and can be generalized to a wide range of spiking neuron models. To distinguish between variants, we use the prefixes "HomNeu" and "HetNeu" to denote homogeneous and heterogeneous configurations at the neuron level, respectively. Under this notation, HetSynLIF can reduce to vanilla LIF neurons (HomNeuLIF), neurons with threshold adaptation (HomNeuALIF), and neuron-level heterogeneous models (HetNeuLIF). This flexibility stems from the incorporation of synapse-specific time constants, which enable HetSynLIF to modulate temporal dynamics at a finer granularity. By analyzing the synaptic dynamics from a presynaptic neuron i to a postsynaptic neuron j, we demonstrate how HetSynLIF generalizes to other neuron models under specific conditions. See Appendix A for detailed proof.

Proposition 1 *If all synaptic decay factors r_{ji} and the reset current decay factor κ_j are identical and equal to a shared value ρ, i.e., $r_{ji} = \kappa_j = \rho$ for all i, j, then the HetSynLIF model reduces to the HomNeuLIF model. Under this condition, the membrane potential of the postsynaptic neuron j at time t in HetSynLIF simplifies to:*

$$V_j^t = \rho \cdot V_j^{t-1} + \sum_i w_{ji} \cdot z_i^t - \vartheta \cdot z_j^{t-1} \tag{10}$$

The resulting equation matches Eq. (3), which suggests that the vanilla LIF model emerges as a special case of HetSynLIF when all synaptic and reset decay factors are identical.

Proposition 2 *Based on Proposition 1, if we further decompose the reset current into a standard component and an additional adaptation current triggered by spikes—characterized by a decay factor ρ_a and a scaling coefficient a—then HetSynLIF generalizes to the HomNeuALIF model. Under this condition, the membrane potential of postsynaptic neuron j at time t in HetSynLIF can be transformed into:*

$$V_j^t = \rho \cdot V_j^{t-1} + \sum_i w_{ji} \cdot z_i^t - J_a^t - \vartheta \cdot z_j^{t-1} \tag{11}$$

This indicates that HetSynLIF effectively reproduces the adaptive dynamics of the HomNeuALIF model proposed by [41] through a structural modification of the reset current, demonstrating its flexibility in modeling neuron dynamics beyond fixed-threshold designs.

Proposition 3 *For each postsynaptic neuron j, if all synaptic decay factors r_{ji} and the reset current decay factor κ_j are identical and equal to a neuron-specific membrane decay factor ρ_j, i.e., $r_{ji} = \kappa_j = \rho_j$ for all i, then HetSynLIF generalizes to the HetNeuLIF model. Under this condition, the membrane potential of postsynaptic neuron j at time t in HetSynLIF simplifies to:*

$$V_j^t = \rho_j \cdot V_j^{t-1} + \sum_i w_{ji} \cdot z_i^t - \vartheta \cdot z_j^{t-1} \tag{12}$$

In this case, each postsynaptic neuron j evolves according to its own membrane decay factor ρ_j, which reflects neuron-level heterogeneity[22]. This shows that HetSynLIF encompasses neuron-level heterogeneity as a special case by assigning shared decay dynamics per neuron.

4 Experiments

As detailed in Section 3.3, HetSynLIF can be reduced to HomNeuLIF, HomNeuALIF, and HetNeuLIF through appropriate parameterization. In this section, we conduct a systematic comparison of the four

variants and further evaluate HetSynLIF against previous state-of-the-art methods. All variants are implemented within our unified framework to ensure theoretical equivalence, controlled ablation, fair comparison, and architectural consistency, enabling us to isolate the specific contribution of synapse-level heterogeneity. For clarity, we prefix feedforward and recurrent SNN implementations with "F-" and "R-", respectively. Further experimental and training details are provided in the Appendix.

4.1 Versatile Timescale Integration in Pattern Generation Task

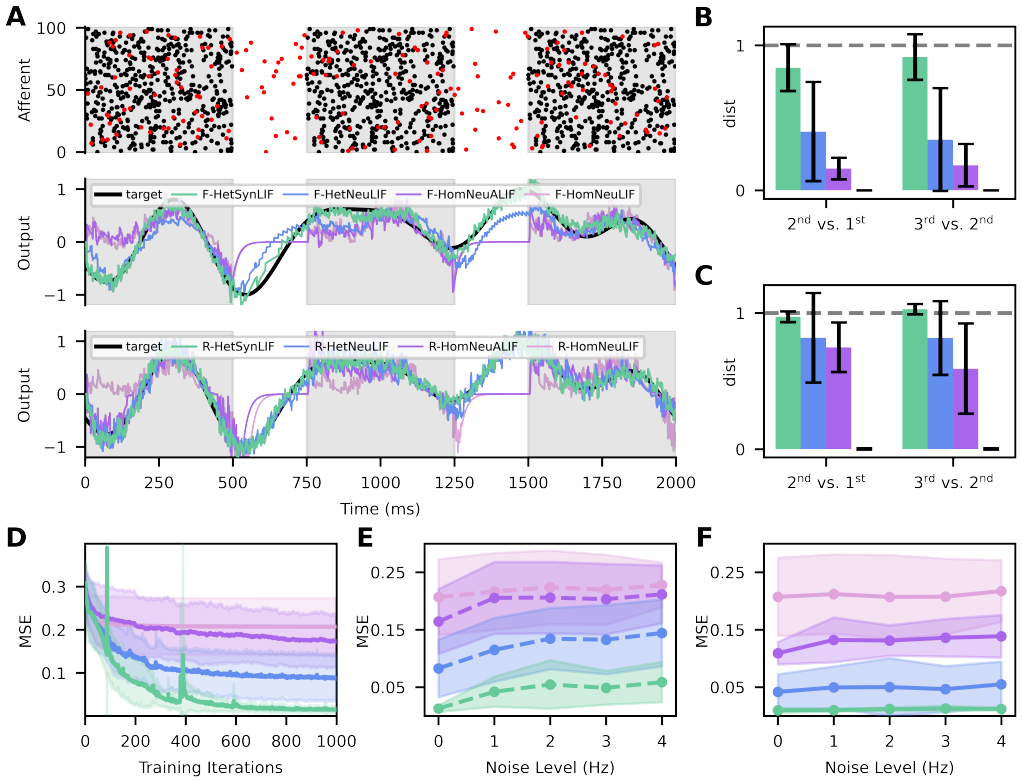

Figure 2: Pattern generation task. (A) An input spike pattern with a fixed Poisson-generated template (10 Hz, black dots) and superimposed noise spikes (2 Hz, red dots). Repeating features are marked by shaded gray regions (Top). Output traces of four FSNN variants (middle) and four RSNN variants (bottom) in response to the same input pattern. (B-C) Normalized distances between output traces at the 2^{nd} vs. 1^{st} and 3^{rd} vs. 2^{nd} occurrences of the repeating input segments for FSNNs (B) and RSNNs (C). Higher means better; see Appendix B. (D) Mean squared error over training iterations for four FSNN variants. (E-F) Mean squared error of FSNN variants (E) and RSNN variants (F) under different noise levels. Data in **B**-**F** are averaged over 10 runs and reported as mean $\pm$ s.d.

We first evaluate HetSynLIF and three variants using one-layer FSNN and RSNN architectures on a more complex version of the pattern generation task, compared to the setups used in previous works [13, 14], where the network is trained to reproduce a target trace in response to structured input spike patterns that consist of repeated segments embedded in Poisson noise, with no input provided between the segments (Fig. 2A; see Appendix for details). We reveal that our HetSynLIF exhibits the fastest convergence and lowest mean squared error (MSE). In contrast, both HomNeuLIF and HomNeuALIF produce near-zero outputs during the input-free intervals between repeated segments, regardless of whether the network is an FSNN or an RSNN. This suggests their limited ability to learn time constants long enough to bridge these temporal gaps (Fig. 2A, D and Appendix Fig. A1). We then compute normalized distances between outputs corresponding to repeated input segments (Fig. 2B for FSNNs, Fig. 2C for RSNNs, see Appendix B), where larger distances indicate a stronger ability to generate target-aligned outputs for identical input segments. HomNeuLIF

shows consistently low distances, indicating nearly identical responses, while HomNeuALIF and HetNeuLIF show slight improvements in FSNNs and modest gains in RSNNs, benefiting from recurrent connections. Nevertheless, all three are inferior to HetSynLIF in performance, which consistently achieves the highest distances across both architectures. Additionally, under varying levels of input noise, HetSynLIF maintains the lowest MSE, while other models exhibit greater degradation (Fig. 2E, F). Finally, we show that HetSynLIF is also capable of generating multiple target patterns simultaneously (Appendix Fig. A2).

4.2 Versatile Timescale Integration in Delayed Match-to-Sample Task

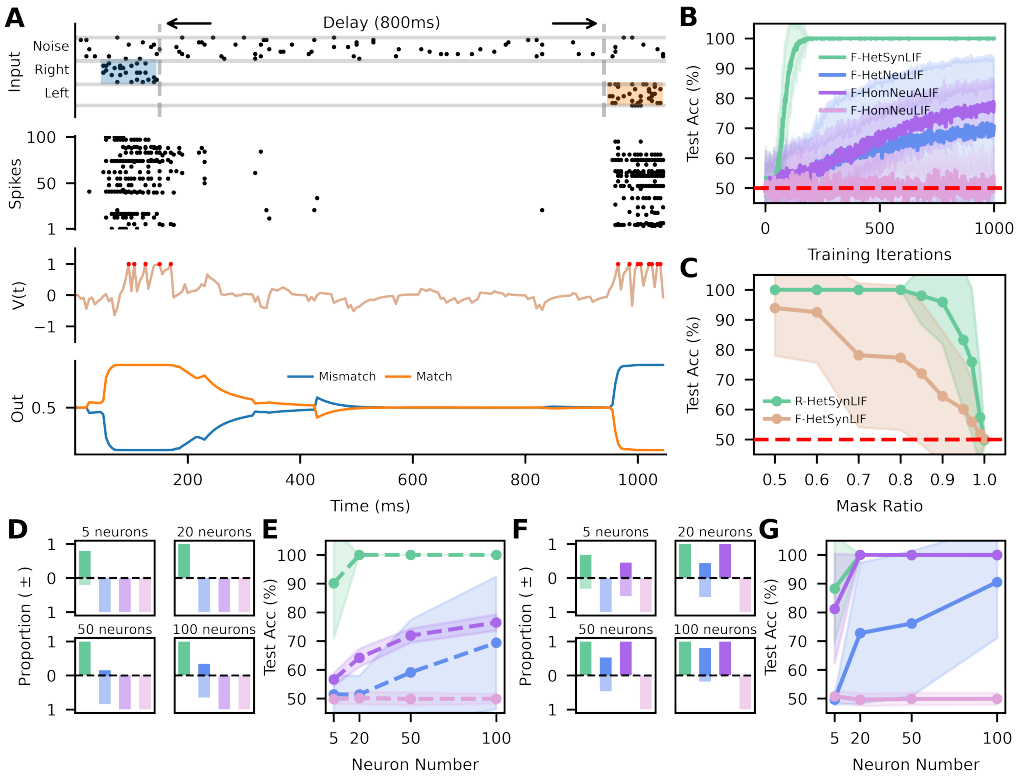

Figure 3: Delayed match-to-sample task. (A) Schematic diagram of a sample mismatch trial with two cues separated by an 800 ms delay (Top). Spiking activity of hidden layer neurons (Second). Membrane potential and spike activities (red dots) of a sample neuron (Third). Softmax outputs of neurons representing match and mismatch (Bottom). (B) Learning curves of four FSNN variants with 100 hidden neurons. (C) Task accuracy under different synaptic time constant masking ratios. (D) Success and failure rates across neuron numbers and FSNN variants. Each grid cell corresponds to a different hidden neuron count (5, 20, 50, 100), containing four centered bars for the four FSNN variants. Each bar is split vertically into a success segment (opaque, above) and a failure segment (transparent, below); trials with accuracy $\geq 90\%$ are considered successful. **D**-**G** follow the same color scheme as **B**. (E) Overall task accuracy of four FSNN variants under varying neuron counts. (F-G) Same as **D**-**E**, but for RSNN variants. Data in **B**-**G** are averaged over 50 runs and reported as mean $\pm$ s.d.

To assess the ability of HetSynLIF to capture multi-timescale dependencies essential for working memory, we adopt the delayed match-to-sample task—a widely used paradigm in neuroscience [42, 43, 44, 45]. The network must judge whether two temporally separated cues belong to the same category (i.e., left or right), requiring short-term processing at cue onset and long-term retention across the delay (Fig. 3A; see Appendix for details). As in the pattern generation task, we compare the performance of eight variants—four FSNNs and four RSNNs—across multiple metrics. Fig. 3B

presents the learning curves of the four FSNN variants, showing that F-HetSynLIF converges fastest and achieves the highest accuracy among FSNN variants. To assess the functional importance of synaptic heterogeneity, we progressively mask a fraction of the synaptic time constants in R-HetSynLIF, rendering them non-trainable. Results show that even with 80% of synapses masked, the model maintains 100% accuracy, highlighting the critical role of heterogeneous temporal integration in sustaining memory over delays (Fig. 3C). We further analyze how performance scales with network size (Fig. 3D–G). Both F-HetSynLIF and R-HetSynLIF consistently achieve higher success rates (defined as accuracy $\geq 90\%$) across varying numbers of hidden neurons, remaining effective even with five neurons, which highlights computational efficiency of synaptic heterogeneity in resource-constrained scenarios. Besides, R-HetSynLIF maintains 100% accuracy even at a 2500 ms delay, with no notable increase in iterations to reach 95% accuracy, and retains nearly 80% accuracy under noise of 20 Hz, demonstrating both stable training efficiency and strong robustness (Appendix Fig. A3).

4.3 Versatile Timescale Integration in Speech Recognition

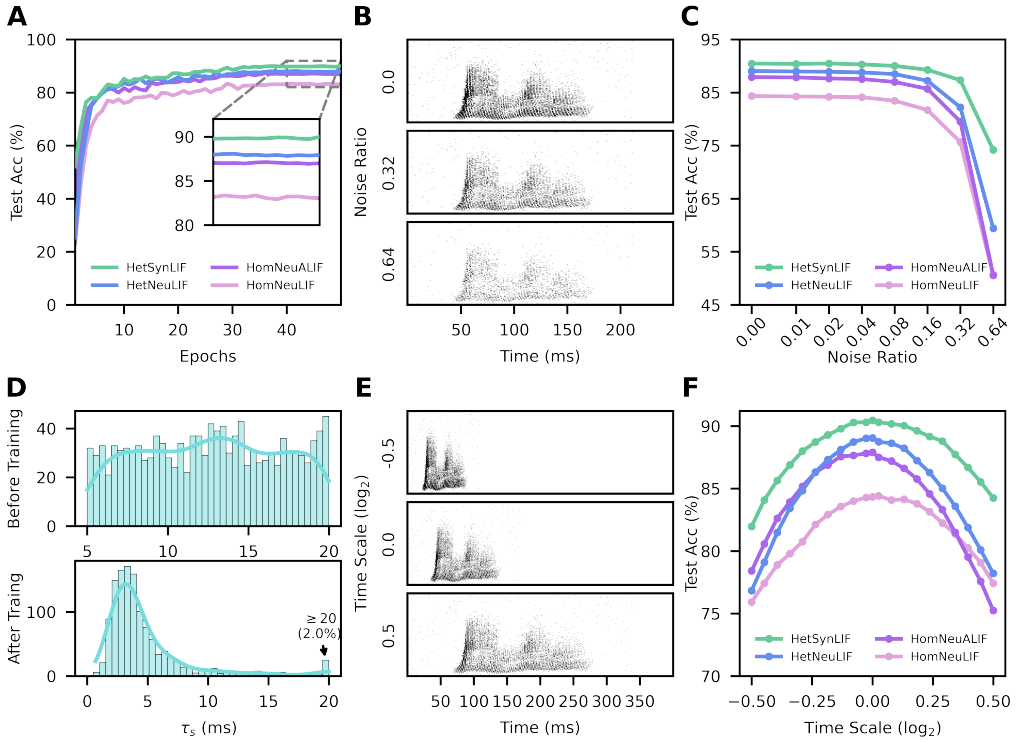

Figure 4: Speech recognition and robustness analysis. (A) Learning curves of four RSNN variants on the SHD dataset. (B) Spike inputs under deletion noise ratios of 0.00, 0.32, and 0.64 (top to bottom), where each ratio denotes the probability of deleting a spike. (C) Test accuracy under varying deletion noise ratios. (D) Distribution of synaptic time constants for connections from the last hidden layer to the output layer, before (top) and after (bottom) training. (E) Spike inputs under time warp conditions with $\log_2$-scaled factors of -0.5 (compressed), 0.0 (original), and 0.5 (stretched), top to bottom. (F) Test accuracy under $\log_2$-scaled time warp factors. Data in **A**, **C**, and **F** are averaged over 10 runs.

We then evaluate the speech recognition capability of four RSNN variants mentioned above on the SHD[46] dataset, using a two-layer recurrent architecture. As shown in Fig. 4A, HetSynLIF consistently achieves the highest accuracy and fastest convergence, followed by HetNeuLIF, HomNeuALIF, and HomNeuLIF. This result underscores the superior modeling capacity of synapse heterogeneity. While neuron heterogeneity also improves performance, assigning distinct time constants to individual synapses offers finer temporal resolution. In contrast, threshold adaptation yields limited gains and remains less effective than neuron heterogeneity. To understand how temporal dynamics evolve during learning, we analyze the distribution of τ_s in the final hidden-to-output layer (Fig. 4D).

Initialized from $\mathcal{U}(5, 20)$ ms, τ_s shifts toward lower values, with 2.0% exceeding 20 ms, forming a mild long-tail that may support slow-varying temporal encoding, loosely resembling biological patterns in Fig. 1A. Similar trends are observed across other layers (Appendix Fig. A5). We further assess robustness under deletion noise and time warp, both of which are introduced only during testing (i.e., trained on clean data and tested with deletion noise or time-warped inputs). In deletion noise, each spike is independently removed with a fixed probability (Fig. 4B). Although performance degrades with increasing noise, the accuracy ranking remains unchanged (Fig. 4C). Notably, when the noise ratio increases from 0.32 to 0.64, HetSynLIF maintains relatively stable, while others drop sharply, demonstrating the strong resilience of synapse heterogeneity to input disruptions. A similar trend holds under time warp distortions (Fig. 4E–F), where HetSynLIF consistently achieves higher accuracy across varying timescales (see Appendix D and Fig. A4 for more details).

4.4 Comparison with Existing Works

We compare our proposed HetSynLIF model with other existing works on the SHD, S-MNIST [47], TiDigits [48], and Ti46-Alpha [49] datasets, and report the results in Table 1. Notably, our two-layer RSNN architecture achieves 92.36% accuracy on the SHD dataset, which, to the best of our knowledge, is the highest reported accuracy among models with similar architectures. In addition, the HetSynLIF model demonstrates outstanding performance on the other three datasets as well, consistently outperforming prior methods, highlighting its effectiveness in processing multi-timescale temporal dynamics across both speech and visual recognition tasks.

Table 1: Accuracy comparison on SHD, S-MNIST, TiDigits, and Ti46-Alpha datasets

Dataset	Method	Acc	Dataset	Method	Acc
SHD	DH-SFNN$^{\text{Nat.Commun24}}$[23]	92.1	TiDigits	BPT-SNN[50]	98.1
	DH-SRNN$^{\text{Nat.Commun24}}$[23]	91.34		M-STIP$^{\text{TNNLS22}}$[51]	98.1
	SRNN$^{\text{ICLR25}}$[52]	91.19		MPD-AL$^{\text{AAAI19}}$[53]	97.5
	SRNN$^{\text{NMI21}}$[54]	90.4		BAE-MPDAL[55]	97.4
	NeuHet-SRNN$^{\text{NatCom21}}$[22]	82.7		TDP-DL[56]	97.16
	SRNN$^{\text{PNAS22}}$[57]	81.6		PBSNLR-DW[58]	96.5
	HetSynLIF (Ours)	**92.36**		**HetSynLIF (Ours)**	**98.99**
S-MNIST	DH-SRNN$^{\text{Nat.Commun24}}$[23]	98.87	Ti46	RSNN[59]	96.44
	SRNN$^{\text{NMI21}}$[54]	98.7		ScSr-SNNs[60]	95.9
	LSTM$^{\text{ICML16}}$[61]	98.2		RSNN$^{\text{NeurIPS19}}$[62]	93.5
	LSNN$^{\text{NeurIPS18}}$[12]	96.4		LSM[63]	92.3
	AHP-SNN$^{\text{NMI22}}$[41]	96.0		S-MLP$^{\text{NeurIPS18}}$[64]	90.98
	HetSynLIF (Ours)	**98.93**		**HetSynLIF (Ours)**	**96.53**

5 Conclusion

In this paper, we propose HetSyn, the first modeling framework with synaptic heterogeneity in SNNs that replaces the conventional membrane time constant with synapse-specific time constants. This design enables versatile timescale integration at the synapse level, grounded in biological observations of synaptic diversity. We instantiate HetSyn as HetSynLIF and theoretically show its generalization ability to existing models. Through comprehensive experiments, we show that HetSynLIF consistently achieves high accuracy on tasks requiring versatile timescale integration, exhibits strong robustness to noise, and maintains computational efficiency under resource constraints. Furthermore, the learned synaptic time constants exhibit a mildly long-tailed distribution, loosely resembling biological observations and reflecting adaptation to multi-timescale processing. These findings underscore the critical role of synaptic heterogeneity in SNNs, not only in enhancing computational performance, but also in aligning with biological principles. By establishing synapse-level modeling as a viable direction, our work points toward promising avenues for advancing brain-inspired learning systems.

Future work could explore the combination of HetSyn with other advanced spiking architectures or training strategies for further improvements in versatile timescale integration and task performance.

Acknowledgments and Disclosure of Funding

This work was in part supported by the Brain Science and Brain-like Intelligence Technology - National Science and Technology Major Project (Grant No. 2025ZD0215600), the National Natural Science Foundation of China (Grant Nos. 92370103 and 62176179), and in part by the Xiaomi Foundation. We thank the reviewers for their insightful and supportive comments.

References

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

[2] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in neuroscience*, 12:409662, 2018.

[3] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[4] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1311–1318, 2019.

[5] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

[6] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11270–11277, 2020.

[7] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36:64043–64058, 2023.

[9] Chaoteng Duan, Jianhao Ding, Shiyan Chen, Zhaofei Yu, and Tiejun Huang. Temporal effective batch normalization in spiking neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 34377–34390. Curran Associates, Inc., 2022.

[10] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

[11] Zuzanna Brzosko, Susanna B Mierau, and Ole Paulsen. Neuromodulation of spike-timing-dependent plasticity: past, present, and future. *Neuron*, 103(4):563–581, 2019.

[12] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

[13] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):3625, 2020.

[14] Yuhan Helena Liu, Stephen Smith, Stefan Mihalas, Eric Shea-Brown, and Uygar Sümbül. Cell-type–specific neuromodulation guides synaptic credit assignment in a spiking neural network. *Proceedings of the National Academy of Sciences*, 118(51):e2111821118, 2021.

[15] Tielin Zhang, Xiang Cheng, Shuncheng Jia, Chengyu T Li, Mu-ming Poo, and Bo Xu. A brain-inspired algorithm that mitigates catastrophic forgetting of artificial and spiking neural networks with low computational cost. *Science Advances*, 9(34):eadi2947, 2023.

[16] Yujie Wu, Rong Zhao, Jun Zhu, Feng Chen, Mingkun Xu, Guoqi Li, Sen Song, Lei Deng, Guanrui Wang, Hao Zheng, et al. Brain-inspired global-local learning incorporated with neuromorphic computing. *Nature Communications*, 13(1):65, 2022.

[17] Uri Hasson, Eunice Yang, Ignacio Vallines, David J Heeger, and Nava Rubin. A hierarchy of temporal receptive windows in human cortex. *Journal of neuroscience*, 28(10):2539–2550, 2008.

[18] Dean V Buonomano and Wolfgang Maass. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–125, 2009.

[19] Stefan J Kiebel, Jean Daunizeau, and Karl J Friston. A hierarchy of time-scales and the brain. *PLoS computational biology*, 4(11):e1000209, 2008.

[20] Alan Baddeley. Working memory: looking back and looking forward. *Nature reviews neuroscience*, 4(10):829–839, 2003.

[21] 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.

[22] Nicolas Perez-Nieves, Vincent CH Leung, Pier Luigi Dragotti, and Dan FM Goodman. Neural heterogeneity promotes robust learning. *Nature communications*, 12(1):5791, 2021.

[23] Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024.

[24] Ann Marie Craig and Hélène Boudin. Molecular heterogeneity of central synapses: afferent and target regulation. *Nature neuroscience*, 4(6):569–578, 2001.

[25] Johanna M Montgomery and Daniel V Madison. State-dependent heterogeneity in synaptic depression between pyramidal cell pairs. *Neuron*, 33(5):765–777, 2002.

[26] François P Chabrol, Alexander Arenz, Martin T Wiechert, Troy W Margrie, and David A DiGregorio. Synaptic diversity enables temporal coding of coincident multisensory inputs in single neurons. *Nature neuroscience*, 18(5):718–727, 2015.

[27] Katie C Bittner, Aaron D Milstein, Christine Grienberger, Sandro Romani, and Jeffrey C Magee. Behavioral time scale synaptic plasticity underlies ca1 place fields. *Science*, 357(6355):1033–1036, 2017.

[28] Luke Campagnola, Stephanie C Seeman, Thomas Chartrand, Lisa Kim, Alex Hoggarth, Clare Gamlin, Shinya Ito, Jessica Trinh, Pasha Davoudian, Cristina Radaelli, et al. Local connectivity and synaptic dynamics in mouse and human neocortex. *Science*, 375(6585):eabj5861, 2022.

[29] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113:54–66, 2015.

[30] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13558–13567, 2020.

[31] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International conference on machine learning*, pages 6316–6325. PMLR, 2021.

[32] Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*, 2023.

[33] Feifan Zhou, Mingqian Fu, Yanxiu Gao, Bo Wang, and Qiang Yu. Rethinking spikes in spiking neural networks for performance enhancement. In *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, pages 374–379. IEEE, 2024.

[34] Sangwoo Hwang, Seunghyun Lee, Dahoon Park, Donghun Lee, and Jaeha Kung. Spikedattention: Training-free and fully spike-driven transformer-to-snn conversion with winner-oriented spike shift for softmax operation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 67422–67445. Curran Associates, Inc., 2024.

[35] 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.

[36] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

[37] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[38] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Yisen Wang, and Zhouchen Lin. Training feedback spiking neural networks by implicit differentiation on the equilibrium state. *Advances in neural information processing systems*, 34:14516–14528, 2021.

[39] 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 qk attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024.

[40] Biswadeep Chakraborty and Saibal Mukhopadhyay. Heterogeneous neuronal and synaptic dynamics for spike-efficient unsupervised learning: Theory and design principles. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] Arjun Rao, Philipp Plank, Andreas Wild, and Wolfgang Maass. A long short-term memory for ai applications in spike-based neuromorphic hardware. *Nature Machine Intelligence*, 4(5):467–479, 2022.

[42] Earl K Miller and Jonathan D Cohen. An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24(1):167–202, 2001.

[43] Ranulfo Romo and Emilio Salinas. Flutter discrimination: neural codes, perception, memory and decision making. *Nature Reviews Neuroscience*, 4(3):203–218, 2003.

[44] Rishidev Chaudhuri, Kenneth Knoblauch, Marie-Alice Gariel, Henry Kennedy, and Xiao-Jing Wang. A large-scale circuit mechanism for hierarchical dynamical processing in the primate cortex. *Neuron*, 88(2):419–431, 2015.

[45] Nuttida Rungratsameetaweemana, Robert Kim, Thiparat Chotibut, and Terrence J Sejnowski. Random noise promotes slow heterogeneous synaptic dynamics important for robust working memory computation. *Proceedings of the National Academy of Sciences*, 122(3):e2316745122, 2025.

[46] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2020.

[47] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[48] R Gary Leonard and George Doddington. Tidigits speech corpus. *Texas Instruments, Inc*, 1993.

[49] M Liberman, R Amsler, K Church, E Fox, C Hafner, J Klavans, et al. Ti46-word speaker-dependent isolated word corpus (ti46). *Philadelphia, PA: Linguistic Data Consortium*, 1993.

[50] Xiaocui Lin, Jiangrong Shen, Jun Wen, and Huajin Tang. Bipolar population threshold encoding for audio recognition with deep spiking neural networks. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

[51] Xiaoling Luo, Hong Qu, Yuchen Wang, Zhang Yi, Jilun Zhang, and Malu Zhang. Supervised learning in multilayer spiking neural networks with spike temporal error backpropagation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10141–10153, 2022.

[52] Yongqi Ding, Lin Zuo, Mengmeng Jing, Pei He, and Hanpu Deng. Rethinking spiking neural networks from an ensemble learning perspective. In *The Thirteenth International Conference on Learning Representations*, 2025.

[53] Malu Zhang, Jibin Wu, Yansong Chua, Xiaoling Luo, Zihan Pan, Dan Liu, and Haizhou Li. Mpd-al: an efficient membrane potential driven aggregate-label learning algorithm for spiking neurons. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1327–1334, 2019.

[54] 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.

[55] Zihan Pan, Yansong Chua, Jibin Wu, Malu Zhang, Haizhou Li, and Eliathamby Ambikairajah. An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks. *Frontiers in neuroscience*, 13:1420, 2020.

[56] Qiang Yu, Jialu Gao, Jianguo Wei, Jing Li, Kay Chen Tan, and Tiejun Huang. Improving multispike learning with plastic synaptic delays. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10254–10265, 2023.

[57] Benjamin Cramer, Sebastian Billaudelle, Simeon Kanya, Aron Leibfried, Andreas Grübl, Vitali Karasenko, Christian Pehle, Korbinian Schreiber, Yannik Stradmann, Johannes Weis, et al. Surrogate gradients for analog neuromorphic computing. *Proceedings of the National Academy of Sciences*, 119(4):e2109194119, 2022.

[58] Malu Zhang, Jibin Wu, Ammar Belatreche, Zihan Pan, Xiurui Xie, Yansong Chua, Guoqi Li, Hong Qu, and Haizhou Li. Supervised learning in spiking neural networks with synaptic delay-weight plasticity. *Neurocomputing*, 409:103–118, 2020.

[59] Wenrui Zhang, Hejia Geng, and Peng Li. Composing recurrent spiking neural networks using locally-recurrent motifs and risk-mitigating architectural optimization. *Frontiers in Neuroscience*, 18:1412559, 2024.

[60] Wenrui Zhang and Peng Li. Skip-connected self-recurrent spiking neural networks with joint intrinsic parameter and synaptic weight training. *Neural computation*, 33(7):1886–1913, 2021.

[61] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pages 1120–1128. PMLR, 2016.

[62] Wenrui Zhang and Peng Li. Spike-train level backpropagation for training deep recurrent spiking neural networks. *Advances in neural information processing systems*, 32, 2019.

[63] Yong Zhang, Peng Li, Yingyezhe Jin, and Yoonsuck Choe. A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE transactions on neural networks and learning systems*, 26(11):2635–2649, 2015.

[64] Yingyezhe Jin, Wenrui Zhang, and Peng Li. Hybrid macro/micro level backpropagation for training deep spiking neural networks. *Advances in neural information processing systems*, 31, 2018.

[65] Robert Gütig. Spiking neurons can discover predictive features by aggregate-label learning. *Science*, 351(6277):aab4113, 2016.

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 clarified our claims in the abstract and introduction.

   Guidelines:

   - The answer NA 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 NA 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 have discussed the limitations of our work in Section 5 and Appendix H.

   Guidelines:

   - The answer NA 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 have provided the full set of assumptions and a complete (and correct) proof in the Method section and Appendix A.

Guidelines:

- The answer NA 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: In Appendix B, C, D, and E, we provide comprehensive details of each experiment, including data processing steps, hyperparameter configurations, and training algorithm. We have submitted our source code in Supplementary Material and we will upload our code to Github upon acceptance.

Guidelines:

- The answer NA 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: Code available at: https://github.com/dzcgood/HetSyn.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided our training and test details in Appendix B, C, D and E.

   Guidelines:

   - The answer NA 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: Our reported results are averaged over several runs. See Figs. 2, 3, 4, A3 and A4.

   Guidelines:

   - The answer NA 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 have provided the computer resource in Appendix E.

   Guidelines:

   - The answer NA 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: The research is in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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 both potential positive societal impacts and negative societal impacts of the work in Appendix H.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: All datasets used in this study have been properly cited; see Section 4.4 and Appendix E for details.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We have submitted our source code used for training and evaluating our models in the Supplementary Material.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

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

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Theoretical Proof

Proposition 1 *If all synaptic decay factors r_{ji} and the reset current decay factor κ_j are identical and equal to a shared value ρ, i.e., $r_{ji} = \kappa_j = \rho$ for all i, j, then the HetSynLIF model reduces to the HomNeuLIF model.*

Proof From Eq. (5) and Eq. (8–9), the membrane potential V_j^t of postsynaptic neuron j at time t in the HetSynLIF model is given by:

$$V_j^t = \left(\sum_i r_{ji} \cdot I_{ji}^{t-1} + \sum_i w_{ji} \cdot z_i^t \right) - \left(\kappa_j \cdot J_j^{t-1} + \vartheta \cdot z_j^{t-1} \right) \tag{13}$$

Under the condition that $r_{ji} = \kappa_j = \rho$, Eq. (13) simplifies to:

$$\begin{aligned} V_j^t &= \left(\sum_i \rho \cdot I_{ji}^{t-1} + \sum_i w_{ji} \cdot z_i^t \right) - \left(\rho \cdot J_j^{t-1} + \vartheta \cdot z_j^{t-1} \right) \\ &= \rho \cdot \left(\sum_i I_{ji}^{t-1} - J_j^{t-1} \right) + \sum_i w_{ji} \cdot z_i^t - \vartheta \cdot z_j^{t-1} \end{aligned} \tag{14}$$

Observe that the term $\sum_i I_{ji}^{t-1} - J_j^{t-1}$ corresponds precisely to the membrane potential at the previous time step $t-1$, as defined in Eq. (8):

$$V_j^{t-1} = \sum_i I_{ji}^{t-1} - J_j^{t-1} \tag{15}$$

Substituting Eq. (14) yields:

$$V_j^t = \rho \cdot V_j^{t-1} + \sum_i w_{ji} \cdot z_i^t - \vartheta \cdot z_j^{t-1} \tag{16}$$

This recurrence relation is identical to the membrane update rule of the HomNeuLIF model, as defined in Eq. (3). Therefore, under the condition $r_{ji} = \kappa_j = \rho$, the HetSynLIF model reduces exactly to the HomNeuLIF model under the specified condition. This completes the proof.

Proposition 2 *Based on Proposition 1, if we further decompose the reset current into a standard component and an additional adaptation current triggered by spikes—characterized by a decay factor ρ_a and a scaling coefficient a—then HetSynLIF generalizes to the HomNeuALIF model.*

Proof Under the condition that the total reset current in the HetSynLIF model consists of two components: a standard reset current J_ϑ^t and an additional adaptation current J_a^t, such that the membrane potential V_j^t of postsynaptic neuron j at time t satisfies:

$$V_j^t = \sum_i I_{ji}^t - J_\vartheta^t - J_a^t \tag{17}$$

The standard reset current J_ϑ^t follows the dynamics defined in Eq. (9), while the adaptation current J_a^t follows the dynamics introduced in [12]:

$$\frac{dJ_a^t}{dt} = -\frac{J_a^t}{\tau_a} + \sum_j a \cdot \delta\left(t - t_{\mathrm{s}}^j\right) \tag{18}$$

$$J_a^t = \rho_a \cdot J_a^{t-1} + a \cdot z_j^{t-1} \tag{19}$$

where $\rho_a = \exp(-\frac{\Delta t}{\tau_a})$ is the decay factor derived from the adaptation time constant τ_a, and a is the adaptation strength. Substituting Eq. (8–9) into Eq. (17), we obtain:

$$V_j^t = \sum_i \left(r_{ji} \cdot I_i^{t-1} + w_{ji} \cdot z_i^t \right) - \kappa_j \cdot J_\vartheta^{t-1} - \vartheta \cdot z_j^{t-1} - J_a^t \tag{20}$$

Under the condition that $r_{ji} = \kappa_j = \rho$, Eq. (20) simplifies to:

$$\begin{aligned} V_j^t &= \rho \cdot \left(\sum_i I_{ji}^{t-1} - J_\vartheta^{t-1} \right) + \sum_i w_{ji} \cdot z_i^t - J_a^t - \vartheta \cdot z_j^{t-1} \\ &= \rho \cdot V_j^{t-1} + \sum_i w_{ji} \cdot z_i^t - J_a^t - \vartheta \cdot z_j^{t-1} \end{aligned} \tag{21}$$

The recurrence in Eq. (21) is similar as the membrane potential dynamics of the HomNeuALIF model defined in [41]. This completes the proof.

Proposition 3 *For each postsynaptic neuron j, if all synaptic decay factors r_{ji} and the reset current decay factor κ_j are identical and equal to a neuron-specific membrane decay factor ρ_j, i.e., $r_{ji} = \kappa_j = \rho_j$ for all i, then HetSynLIF generalizes to the HetNeuLIF model.*

Proof From Eq. (5) and Eq. (8–9), the membrane potential V_j^t of postsynaptic neuron j in the HetSynLIF model is given by Eq. (13). Under the condition that $r_{ji} = \kappa_j = \rho_j$, substituting this into Eq. (13), we obtain:

$$V_j^t = \left(\sum_i \rho_j \cdot I_{ji}^{t-1} + \sum_i w_{ji} \cdot z_i^t \right) - \left(\rho_j \cdot J_j^{t-1} + \vartheta \cdot z_j^{t-1} \right) \tag{22}$$

Rewriting Eq. (22) by extracting ρ_j as a common factor, we have:

$$V_j^t = \rho_j \cdot \left(\sum_i I_{ji}^{t-1} - J_j^{t-1} \right) + \sum_i w_{ji} \cdot z_i^t - \vartheta \cdot z_j^{t-1} \tag{23}$$

Substituting Eq. (15) into Eq. (23) yields:

$$V_j^t = \rho_j \cdot V_j^{t-1} + \sum_i w_{ji} \cdot z_i^t - \vartheta \cdot z_j^{t-1} \tag{24}$$

This update rule corresponds to the membrane dynamics of the HetNeuLIF model, in which each neuron has its own membrane decay factor ρ_j. Thus, under given condition, the HetSynLIF model reduces to the HetNeuLIF model. This completes the proof.

B Details of the Pattern Generation Task

Task Description In this task, network is trained to reproduce a continuous target trace in response to structured input spike patterns. Each input consists of 100 afferent channels spanning a 2000-ms window. A single 500-ms spike segment is first generated using a 10 Hz Poisson process and then embedded three times within the input window, evenly spaced and separated by 250-ms silent intervals (i.e., periods with no input spikes from the structured segment, though background noise may still occur). To perturb the input structure, global background noise is superimposed across all channels, and regenerated independently at each training iteration. In our experiments (Fig. 2E, F), we evaluate noise levels with rates of 1, 2, 3, and 4 Hz to examine the model's robustness under increasing perturbation. The target trace is constructed as the sum of five cosine components with a base frequency of 0.5 Hz, with each component modulated by a random amplitude sampled from $\mathcal{U}(0, 1)$ and a phase sampled from $\mathcal{U}(0, 2\pi)$. The resulting signal is then normalized to the range $[-1, 1]$.

Loss Function We train the network using a loss that combines a task-specific term and a regularization term. The task loss is defined as the mean squared error (MSE) between the network output (y^t) and the target trace ($y^{*,t}$), averaged over all timesteps. The regularization term penalizes deviations of each neuron's firing rate from a target rate of 10 Hz, computed as the mean squared relative error across neurons. The two terms are weighted using a fixed coefficient of 0.01 for the regularization.

Computation of the Normalized Distances In Fig. 2B and C, we quantify the model's ability to generate temporally distinct outputs for repeated input segments by computing the normalized distance between the output traces corresponding to different occurrences of the same structured input. Specifically, for any two repeated segments (e.g., the first and second), we compute the mean squared difference between their output traces, and normalize it by the corresponding difference between the target traces:

$$\text{dist}(\text{2nd}, \text{1st}) = \frac{\sum_t \left(y^t_{\text{2nd}} - y^t_{\text{1st}}\right)^2}{\sum_t \left(y^{*,t}_{\text{2nd}} - y^{*,t}_{\text{1st}}\right)^2} \tag{25}$$

Here, a higher distance indicates that the model produces differentiated outputs for repeated inputs, suggesting alignment with the target trace rather than generating identical responses to identical stimuli. Conversely, a lower value implies that the model outputs remain nearly the same across repetitions.

Experiment Settings We adopt a 100-100-1 network architecture for all models, comprising 100 input channels, 100 hidden neurons, and a single leaky-integrate readout neuron. The synaptic time constant τ_{s} is fixed at 20 ms for HomNeuLIF and HomNeuALIF, and initialized by sampling from $\mathcal{N}(20, 5)$ ms for HetNeuLIF and HetSynLIF. The reset time constant τ_J is uniformly set to 20 ms across all models. For HomNeuALIF, the adaptation time constant and adaptation strength are set to $\tau_a = 500$ ms and $a = 0.01$, respectively. All decay factors are constrained to the range $[0, 1]$ during training. For the surrogate function, we use a triangular-shaped derivative defined as $\frac{\partial z^t}{\partial V^t} = \gamma \cdot \max\left(0, 1 - \left|\frac{V^t - \vartheta}{\zeta \cdot \vartheta}\right|\right)$, we set $\gamma = 1$ and $\zeta = 1$. Training is performed using the Adam optimizer with a StepLR learning rate scheduler, where the learning rate is initialized at 1e-3 and decayed by a factor of 0.8 every 100 iterations. All models are trained for 1000 iterations with a batch size of 1.

C Details of the Delayed Match-to-Sample Task

Task Description In this task, the network is required to determine whether two temporally separated cues belong to the same category (i.e., left or right). There are four possible cue combinations: `left-left`, `left-right`, `right-left`, and `right-right`. The `left-left` and `right-right` conditions are labeled as "match" , while the remaining two are considered "mismatch". The input consists of 30 channels, divided equally into left-cue, right-cue, and noise channels (10 each). Each cue is delivered through either the left or right group using independent Poisson spike trains at 40 Hz. The first cue begins at 50 ms and lasts for 100 ms, followed by an 800 ms delay. The second cue then appears, also lasting 100 ms. Throughout the entire 1050 ms input window, continuous background noise is added on the noise channels using a 10 Hz Poisson process. To evaluate the model's ability to retain the first cue and compare it to the second after a long delay, we compute the decision by comparing the membrane potentials of two output neurons—corresponding to "match" and "mismatch"—at the final millisecond of the second cue (1050 ms), and the class associated with the higher membrane potential is selected as the predicted label.

Loss Function Similar to the pattern generation task, the network is trained using a loss function composed of a task-specific term and a regularization term. The regularization term remains unchanged and penalizes deviations of each neuron's firing rates from a target of 10 Hz. The task-specific loss is defined as the cross-entropy between the predicted class probabilities and the target labels.

Experiment Settings In this task, the synaptic time constant τ_{s} is fixed at 40 ms for HomNeuLIF and HomNeuALIF, and initialized by sampling from $\mathcal{N}(40, 4)$ ms for HetNeuLIF and HetSynLIF. The reset time constant τ_J is uniformly set to 40 ms across all models. For HomNeuALIF, the adaptation time constant and adaptation strength are set to $\tau_a = 800$ ms and $a = 0.15$, respectively. All decay factors are constrained to the range $[0, 1]$ during training. A simulation timestep of $\Delta t = 5$ ms is used to reduce computational overhead. We use the same surrogate function as in the pattern generation task and set $\gamma = 1$ and $\zeta = 1$. Training is performed using the Adam optimizer with a StepLR

learning rate scheduler, where the learning rate is initialized at 5e-3 and decayed by a factor of 0.8 every 100 iterations. All models are trained for 1000 iterations with a batch size of 32.

D Details of the Speech Recognition Task

SHD Dataset and Preprocessing The SHD dataset comprises 10,420 high-quality audio samples of spoken digits (0-9) in English and German. It includes 12 speakers—6 female and 6 male—aged between 21 and 56, with each speaker contributing approximately 40 utterances per digit for each language. The dataset is divided into training and testing sets, containing 8,156 and 2,264 samples, respectively. Before feeding into neural networks, we first align all audio recordings to a fixed duration of 1000 ms by trimming or zero-padding, and then sample the resulting spike trains using a 4 ms time bin, yielding a 250×700 input matrix per recording (250 timesteps $\times$ 700 input channels).

Deletion Noise Spiking systems deployed in real-world scenarios often face partial signal loss or sensor failures, making robustness to missing inputs a critical property. To assess this capability, we evaluate performance under deletion noise, which simulates missing input events by independently removing each spike with a fixed probability p. Specifically, for each spike input (i.e., binary value 1), it is retained with probability $1 - p$ and set to 0 with probability p. We test seven noise levels with $p \in \{0.0, 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64\}$, corresponding to $p = 0.01 \times 2^n$ for $n = 0$ to 6, plus a clean condition.

Temporal scaling Temporal scaling is a fundamental aspect of sensory processing, as biological systems such as the auditory and motor systems are capable of recognizing patterns across a wide range of speeds. To test whether SNNs equipped with HetSyn exhibit similar robustness, we simulate temporal variability by scaling each spike time with a global warping factor. That is, each spike time is multiplied by a constant factor α, resulting in a globally compressed or stretched spike sequence. Accordingly, the total number of timesteps changes proportionally to the warping factor. We evaluate 20 different warping conditions, with factors sampled as $\log_2 \alpha \sim \mathcal{U}(-0.5, 0.5)$, covering both temporal compression ($\log_2 \alpha < 0$) and dilation ($\log_2 \alpha > 0$) at varying scales.

Experiment Settings As in [23, 54], we adopt leaky-integrate neurons for the readout layer to decode the output of the network, where the predicted possibility for class i is obtained by summing the softmax-normalized membrane potential of the output neuron over time, i.e., $\hat{y}_i = \sum_t \text{softmax}(V_{\text{out}}^t)$. For better performance, we use multi-Gaussian curve[54] as surrogate function, and train the network using the standard cross-entropy loss. The synaptic time constant τ_{s} is fixed at 20 ms for HomNeuLIF and HomNeuALIF, and initialized by sampling from $\mathcal{U}(5, 20)$ ms for HetNeuLIF and HetSynLIF. The reset time constant τ_J is uniformly set to 20 ms across all models. For HomNeuALIF, the adaptation time constant and adaptation strength are set to $\tau_a = 100$ ms and $a = 0.05$, respectively. All decay factors are constrained to the range $[0, 1]$ during training. The model is optimized using the AdamW with a learning rate of 2e-3 and a weight decay of 4e-3. We apply a cosine annealing schedule with 5% warm-up, computed per batch, and fix the learning rate after 40 epochs.

E Datasets and Configurations

SHD Same as Appendix D.

S-MNIST Sequential MNIST (S-MNIST) is a sequential version of the standard MNIST dataset, where each 28×28 image is reshaped into a sequence of 784 inputs. At each timestep, a single pixel is presented to the model in a row-wise scan from top-left to bottom-right. The MNIST dataset consists of grayscale images of handwritten digits (0–9), with 60,000 training and 10,000 test samples. We follow the original train/test split in our evaluation.

TiDigits We use the adults subset of the TiDigits dataset, which comprises isolated utterances of the 11 English digit classes ("zero"–"nine" and "oh"), with standard training and testing splits of 2,464 and 2,486 speech samples. We use threshold crossing encoding method proposed by Gutig [65] to encode TiDigits into spikes, with nfft=512, 16 Mel filters from 360–8000 Hz, and 15 thresholds

per channel for crossing. We use the first 200 timesteps of each sample, yielding an input of shape 200×500 (timesteps $\times$ channels).

Ti46-Alpha Ti46-Alpha, consisting of the 26 English alphabet letters, is a subset of the Ti46-Word dataset. We apply the same encoding method as in TiDigits, yielding input representations of size 200×500.

Experiment settings All experiments are conducted using NVIDIA RTX 4090 and Tesla V100-PCIE-16GB GPUs. We employed a cosine-annealing learning rate scheduler applied at each batch step. For S-MNIST, classification is based on spike counts from the output layer neurons, whereas for all other datasets, predictions are derived from temporally integrated outputs, as previously described for SHD. A detailed overview of the experimental settings and hyperparameter configurations is provided in Table 2.

Table 2: Experiment settings and hyperparameter configurations for different datasets

Dataset	SHD	S-MNIST	TiDigits	Ti46-Alpha
learning rate	2e-3	1e-3	2e-3	4e-3
dropout rate	0.2	0	0.5	0.2
epochs	100	150	50	50
batch size	32	256	32	32
warmup ratio	0.05	0.05	0.05	0.05
optimizer	AdamW	AdamW	AdamW	AdamW
weight decay	4e-3	0	1e-2	4e-3
architecture	SRNN	SRNN	SRNN	SRNN
hidden neuron number	[128, 64]	[64, 64]	[64, 32]	[128, 128]
ϑ	1.0	1.0	1.0	1.0
Δt	1e-3	1e-3	1e-3	1e-3
τ_J	20e-3	20e-3	20e-3	20e-3
initialization of τ_{s}	$\mathcal{U}$(5e-3, 20e-3)	$\mathcal{U}$(5e-3, 20e-3)	$\mathcal{U}$(5e-3, 20e-3)	$\mathcal{U}$(5e-3, 20e-3)

F Model Complexity Analysis

We conduct a comparative analysis of model parameters across several representative SNN models with distinct neuronal dynamics. The models considered include four FSNN variants, four RSNN variants, and two dendritic heterogeneity models (DH-SFNN and DH-SRNN [23]). For clarity, we assume that each layer consists of N neurons and receives M inputs, with D representing the number of dendritic branches per neuron. The theoretical results are summarized in Table 3.

The analysis indicates that HetSynLIF introduces a moderate increase in trainable model parameters under identical architectural configurations and neuron counts. Intriguingly, HetSynLIF achieves comparable or even superior performance with significantly fewer neurons and faster convergence (Fig. 2D, 3B, 3D, 4A, and Appendix Fig. A1), suggesting that the increased per-unit complexity is effectively offset and may even constitute an advantage in terms of overall training efficiency. Given the notable performance gains and the promising potential of synaptic heterogeneity in SNNs, the additional per-unit complexity represents a reasonable trade-off at this early stage, which is natural and biologically grounded, as synapses greatly outnumber neurons in the brain. Future studies may further mitigate the computational complexity of HetSynLIF by exploring sparse synaptic connectivity and lightweight architectural designs, as inspired by [12, 22].

Table 3: Comparison of trainable model parameters

Models	**Synaptic Parameters**	**Neuron Parameters**	**Total Parameters**
F-HomNeuLIF	MN	0	MN
F-HetNeuLIF	MN	N	$MN + N$
F-HomNeuALIF	MN	0	MN
F-HetSynLIF	$2MN$	0	$2MN$
DH-SFNN	MN	$N + ND$	$MN + (D + 1)N$
R-HomNeuLIF	$MN + NN$	0	$MN + NN$
R-HetNeuLIF	$MN + NN$	N	$MN + NN + N$
R-HomNeuALIF	$MN + NN$	0	$MN + NN$
R-HetSynLIF	$2MN + 2NN$	0	$2MN + 2NN$
DH-SRNN	$MN + NN$	$N + ND$	$MN + NN + (D + 1)N$

G Evolution of Synaptic Time Constants During Training

The DMS task inherently relies on both short-term and long-term memory capabilities: long-term memory enables the model to retain cue information (e.g., "left" or "right") across the delay period, while short-term memory helps to suppress irrelevant noise. During training, the time constants in the cue-related pathways tend to increase toward the delay duration to sustain relevant information, while those in noise-related pathways decrease to promote rapid forgetting.

To demonstrate this, we initialize the time constants τ from Gaussian distributions with means $\mu \in \{100, 200, 400\}$ ms and standard deviations of 0.1μ. Each configuration is trained for five independent runs, and the averaged results in Table 4 indicate the expected divergence of learned time constants between the cue and noise pathways. In addition, approximately 5–6% of synapses develop time constants comparable to the task delay, and resetting these synapses to their initial mean values leads to a substantial decrease in accuracy. These results demonstrate the task-aligned functional specialization of synaptic time constants, and further support the effectiveness of HetSyn in versatile timescale integration by adaptively tuning synaptic dynamics to input characteristics during training.

Table 4: Analysis of time constant statistics

Initialization	$\mathcal{N}(100, 10^2)$	$\mathcal{N}(200, 20^2)$	$\mathcal{N}(400, 40^2)$
Cue-channel (after training)	154.38 (+54.38%)	252.43 (+26.22%)	430.70 (+7.68%)
Noise-channel (after training)	85.59 (-14.41%)	181.93 (-9.04%)	382.29 (-4.43%)
Synapses with $\tau > 700$ ms	5.25%	6.05%	5.80%
Accuracy	99.97%	100.00%	99.87%
Accuracy (reset long τ)	75.68% (-24.29%)	76.47% (-23.53%)	77.15% (-22.72%)

H Discussion

Broader impacts By introducing synaptic heterogeneity into SNNs, our work advances the modeling of biologically inspired temporal dynamics and enhances the computational performance of SNNs. This has the potential to improve both learning performance and energy efficiency in neuromorphic computing systems, contributing to the development of low-power, event-driven AI applications. Furthermore, our study is purely computational and does not involve sensitive data or human subjects, and we anticipate no adverse societal or environmental impacts of out study.

Limitations This work provides a step toward understanding the role of synaptic heterogeneity in SNNs while introducing additional per-unit complexity and increased training cost. Future work will explore approaches for reducing computational complexity, such as sparse synaptic connectivity and lightweight architectural designs. Moreover, as this work focuses on a subset of synaptic properties, a broader range of other synaptic features and more sophisticated architectures remain to be explored.

I Supplementary Figures

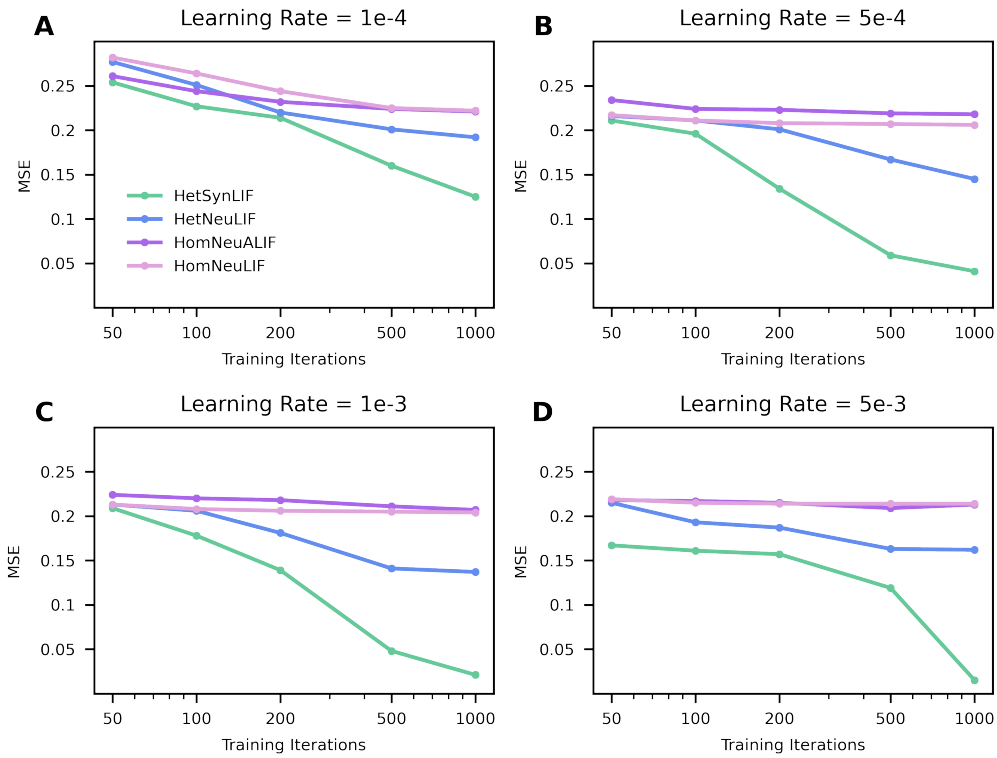

Figure A1: Convergence speed evaluation of four FSNN variants on the pattern generation task, trained with learning rates 1e-4 (A), 5e-4 (B), 1e-3 (C), 5e-3 (D). The curves show the mean squared error (MSE) over training iterations.

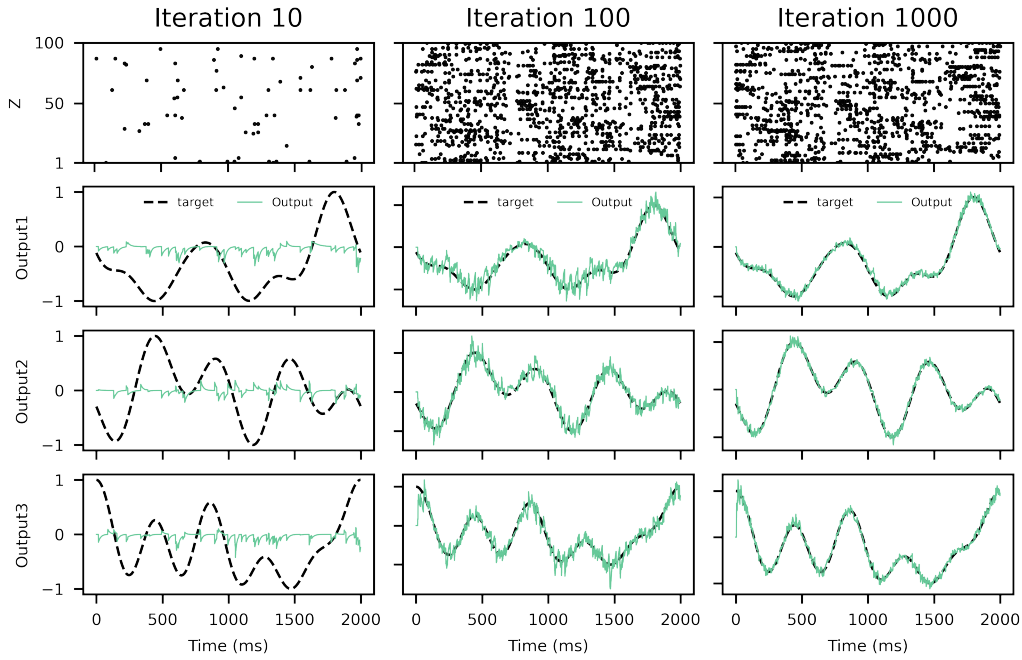

Figure A2: Dynamics of F-HetSynLIF during training on the pattern generation task with simultaneous generation of three output patterns. From left to right: network dynamics at 10, 100, and 1000 training iterations. Each column shows spike activity of hidden neurons (top), followed by predicted traces (colored lines) and targets (black dashed lines) for the three outputs.

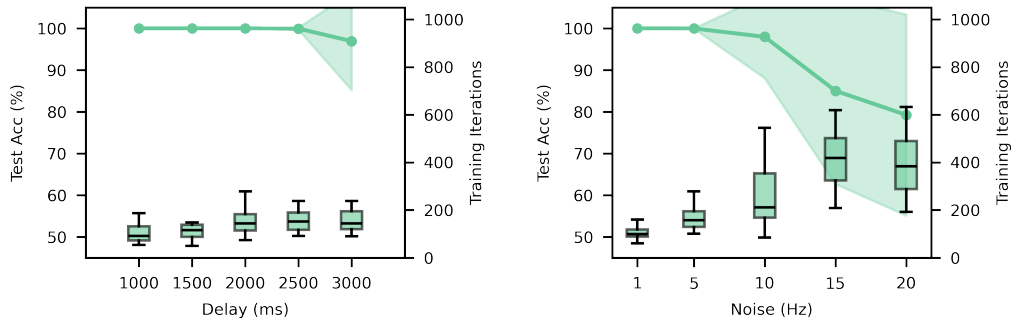

Figure A3: Performance and training efficiency of R-HetSynLIF on the delayed match-to-sample task under varying delays and noise levels. Left: Test accuracy (solid line, left y-axis) and training iterations required to reach 95% accuracy (boxplot, right y-axis; window size = 10) across different delay durations. Right: Same metrics evaluated under varying levels of addition noise (Hz).

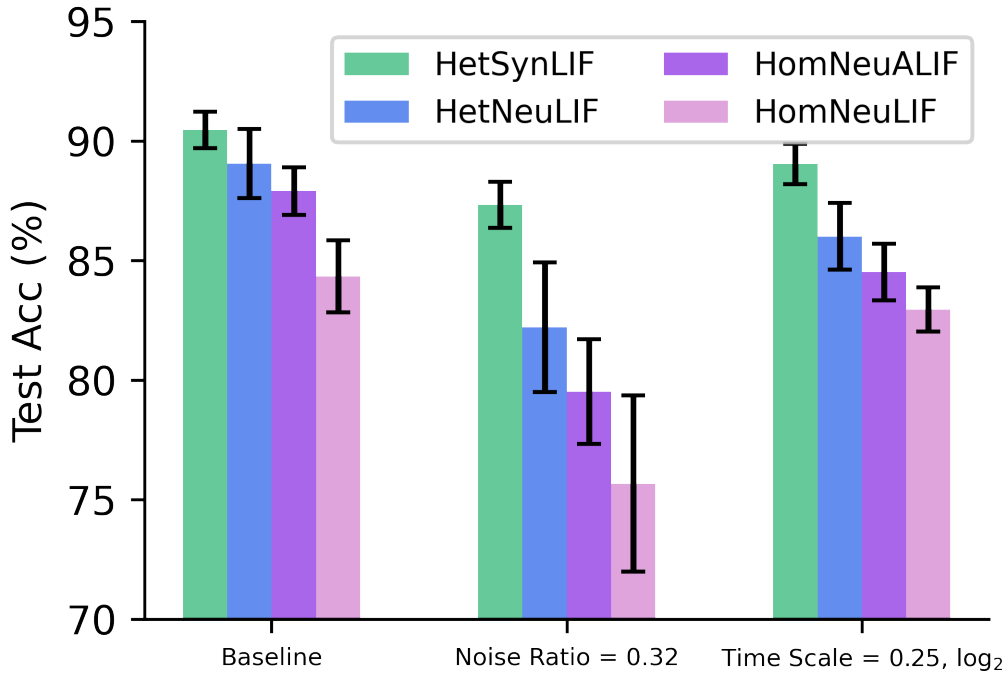

Figure A4: Bar plots show test accuracy of four RSNN variants on the SHD dataset under three conditions: baseline (left), deletion noise ratio 0.32 (middle, from Fig. 4C), and time warp scale 0.25 ($\log_2$, right, from Fig. 4F). Error bars represent standard deviation over 10 runs.

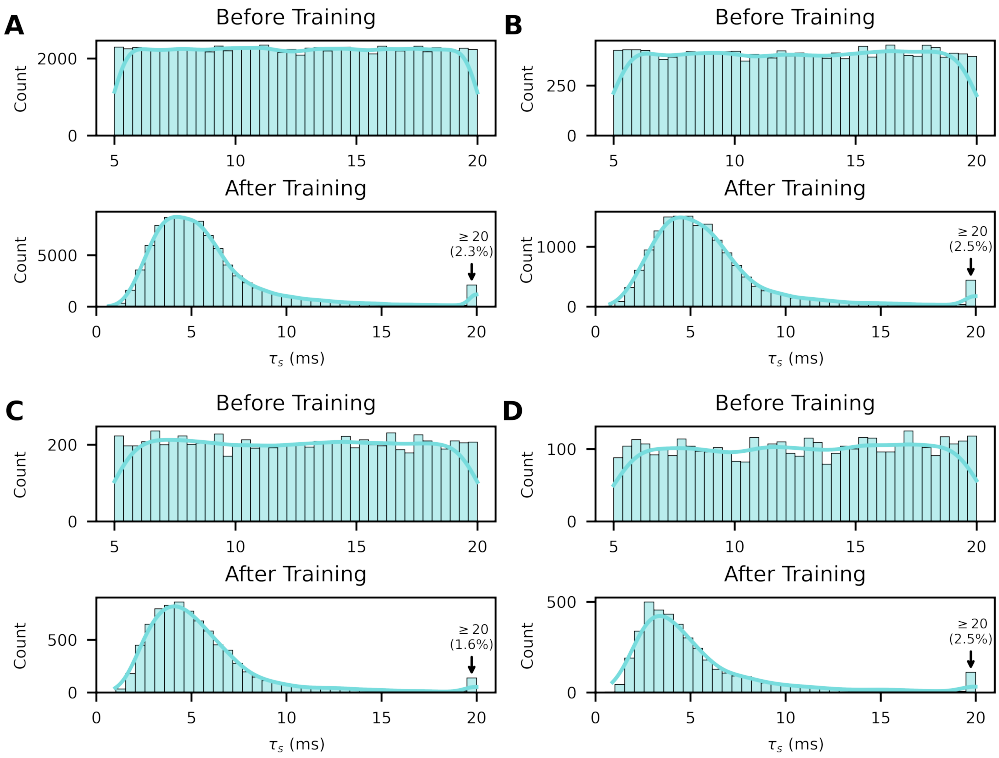

Figure A5: Distributions of synaptic time constants (τ_s) across all connection types in the two-layer R-HetSynLIF. From (A) to (D): input to hidden layer 1, recurrent connections within hidden layer 1, hidden layer 1 to hidden layer 2, and recurrent connections within hidden layer 2. Each subplot shows the distribution of τ_s before training (top) and after training (bottom).