When a Window Is Not an Action: Selective Phase-Script Deliberation for Sliding-Window Human Activity Recognition

Qi Teng, Xue'er Wang*
Zhengzhou University, Sun Yat-sen University
teqi159@gmail.com, wxe0121@hotmail.com

Abstract

Sliding-window human activity recognition (HAR) segments continuous sensor streams into fixed-length, overlapping windows and trains a classifier to assign a single activity label to each window. In practice, window length and stride are dictated by sampling rates, latency or computation constraints, and assumed motion durations, yet they need not align with complete motion cycles or activity transitions. Individual windows may therefore capture only part of a repeated motion, fragments shared by related activities, or mixed evidence near a transition. Yet each window still receives a single label, creating a granularity mismatch between activity-level supervision and partial local motion evidence. To address this mismatch, we propose PSDNet (Phase-Script Deliberation Network), which decouples local phase interpretation from final activity classification. PSDNet encodes the current window into latent phase primitives, integrates them with recent history against learned class-specific phase scripts—compact templates of short-term phase patterns—and produces an initial evidential prediction with an uncertainty score. When uncertainty remains high, a selective deliberation module leverages targeted contextual evidence to refine classification scores over a small candidate set. An auxiliary boundary objective further encourages sensitivity to activity transitions without serving as a hard routing rule. This design concentrates extra computation where direct classification is least reliable while keeping unambiguous cases efficient. Experiments across eight public HAR benchmarks demonstrate that PSDNet consistently improves accuracy and weighted F1 over strong CNN- and sequence-based baselines, with extensive analyses on boundary-adjacent windows and look-alike confusion pairs further validating phase-aware selective deliberation under insufficient local evidence.The source code is publicly available at `https://github.com/WXE0121/PSDNet`.

1 Introduction

Human activity recognition (HAR) from wearable and mobile sensors is most often formulated as a sliding-window classification problem: a continuous multivariate sensor stream is segmented into fixed-length, often overlapping windows, and each window is assigned one activity label Chen et al. [2021]. This formulation is simple, streamable, and easy to optimize. It gives the model a uniform input shape and allows continuous sensor streams to be converted into many labeled training samples, which is why it underlies a large family of CNN-, RNN-, and Transformer-based HAR systems Zhou et al. [2022]. In practice, the window length and stride are usually chosen according to the sensor sampling rate, latency or computation constraints, and empirical assumptions about the duration of characteristic motion units. However, these engineering choices do not guarantee that a window is

*Corresponding author.

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

aligned with a complete motion cycle or with the moment when one activity changes to another Bock et al. [2025].

This mismatch between segmentation and activity structure can make the evidence available in a window incomplete or ambiguous. Within a continuous activity such as walking, a sliding window may capture only the middle or tail of a repeated motion cycle. Around an activity change, a window may contain evidence from both the preceding and following activities. Related activities may also share short local fragments: *upstairs* and *downstairs* may both contain a leg-lifting fragment; *sit-down* and *stand-up* may both include a forward-leaning segment Bock et al. [2024], Geissler et al. [2024]. These cases are difficult not merely because the classifier lacks capacity, but because the granularity of the observation does not match the granularity of the label: the supervision describes an activity-level category, while the current window may expose only partial local evidence that, on its own, is not enough to identify the activity reliably.

This granularity mismatch helps explain a recurring pattern in HAR errors: confusions often concentrate near activity boundaries and among activities that contain similar local motion fragments Xia et al. [2024]. Existing methods commonly respond by strengthening feature extraction, enlarging temporal receptive fields, or aggregating more contextual information Abedin et al. [2021]. These directions are useful, but many of them still preserve the same target abstraction: the current window, possibly enhanced by context, is mapped directly to a single activity label Wang et al. [2024b]. In this formulation, context is often used as a generic feature booster rather than as evidence for judging whether a partial local observation is compatible with a class-specific pattern of phase progression Shao et al. [2024]. Moreover, applying equally heavy temporal reasoning to every window can be inefficient for easy cases and insufficiently targeted for genuinely ambiguous ones.

We take a different view. Instead of treating each window as a self-contained activity instance, we treat it as local phase evidence within an evolving activity sequence. Based on this view, we propose PSDNet (*Phase-Script Deliberation Network*). PSDNet first maps the current window to latent phase primitives, producing a representation better aligned with local motion evidence than an immediate hard activity decision. It then summarizes recent history and evaluates class-specific phase-script compatibility, allowing the model to reason about whether the current local evidence is consistent with plausible activity-level evolution. From this local evidence, PSDNet estimates uncertainty and invokes an additional deliberation step only when the window remains ambiguous. The deliberation module updates the logits over a small candidate set using script-aware contextual evidence, while an auxiliary boundary branch encourages sensitivity to possible activity changes without hard-wiring boundary prediction into the routing mechanism.

Our central claim is that, for sliding-window HAR, ambiguous windows should be classified by combining local motion evidence with recent phase history rather than relying on the current window alone. Our contributions are threefold:

- We formulate ambiguous sliding-window HAR as a granularity-mismatch problem induced by fixed-window segmentation: activity-level labels may supervise windows that contain only partial local motion evidence. This formulation provides a unified explanation for boundary ambiguity and look-alike class confusions.

- We introduce PSDNet, a phase-script deliberation framework that separates phase-level interpretation from activity-level decision making. It encodes each window as latent phase evidence, scores this evidence against class-specific phase scripts using recent history, and activates candidate-focused deliberation through an explicit uncertainty gate when local evidence remains uncertain.

- We demonstrate that, under a unified evaluation protocol, PSDNet attains the best mean accuracy and weighted F1 on all eight public HAR benchmarks among the compared CNN- and sequence-based methods. Ablations, boundary/look-alike diagnostics, and accuracy–efficiency Pareto analyses further show that the gains are driven by phase modeling and uncertainty-gated deliberation rather than simply applying heavier computation to every window.

2 Related Work

Sliding-window and contextual HAR. Sensor-based HAR has progressed from hand-crafted descriptors to deep end-to-end recognition from raw multivariate windows Wang et al. [2019]. CNN models learn local temporal and cross-channel patterns directly from sensor windows Ronao and Cho [2016], while CNN–RNN hybrids such as DeepConvLSTM add recurrent aggregation over learned features Ordóñez and Roggen [2016]. Self-attention and Transformer-style models further enlarge the temporal receptive field Betancourt et al. [2020], Dirgová Luptáková et al. [2022]. A complementary line explicitly incorporates longer temporal structure or dense prediction: P2LHAP casts recognition, segmentation, and forecasting as Patch-to-Label Seq2Seq learning Li et al. [2024]; temporal action localization for inertial HAR performs timeline-level segment reasoning Bock et al. [2024]; and DeepConvContext models intra-window and inter-window dependencies over ordered windows Bock et al. [2025]. These works confirm that stronger encoders and broader context are useful, but they usually preserve the same target abstraction: activity labels or segments are decoded directly from windows or window sequences. Our focus is different. We treat the current window as partial phase evidence and ask whether that evidence is compatible with a class-specific short-horizon evolution.

Transition ambiguity and phase-level representations. Several HAR studies note that recognition errors are concentrated around transitional periods or among activities with locally similar motion patterns Reyes-Ortiz et al. [2016], Chen et al. [2020]. Another related direction decomposes activities into smaller motion units, such as motion primitives, intermediate states, or hierarchical stages. Early wearable HAR used motion-primitive vocabularies and forests Zhang and Sawchuk [2012a], Diep et al. [2016], and recent work such as MoPFormer revisits this idea with learned primitive tokens and Transformer representations Zhang et al. [2025]. PSDNet shares the intuition that local motion units are useful, but differs in how they are used: phase primitives are latent, learned end-to-end, and coupled to class-specific scripts for selective refinement rather than treated as external labels, fixed codebooks, or standalone interpretability targets.

Uncertainty and selective computation. Uncertainty has been used in context-aware activity recognition to improve robustness and detect unknown contexts Huo et al. [2020]. Our use of uncertainty is narrower and more structural. The goal is not merely to estimate confidence after prediction, but to decide when direct local classification is insufficient because the window exposes only incomplete evidence. Thus, selective computation is part of the recognition formulation itself: easy windows are resolved by a phase-aware local predictor, while ambiguous windows receive script-aware deliberation.

3 Method

We design PSDNet around a simple premise: a short sensor window should not always be treated as a self-sufficient action instance. The model first interprets the current window as *local phase evidence*, then uses class-specific short-horizon scripts to form a locally grounded prediction, and finally invokes heavier contextual deliberation only when this local evidence remains uncertain. The main text keeps only the equations needed to define the core predictor and training objective; implementation-level details, notation, and pseudocode are provided in Appendix A.

3.1 Problem Setup

After sliding-window segmentation, the training set contains ordered window sequences

$$\mathcal{D} = \{(\mathbf{X}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}, \quad \mathbf{X}^{(n)} = (\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_{T_n}^{(n)}), \quad \mathbf{y}^{(n)} = (y_1^{(n)}, \ldots, y_{T_n}^{(n)}), \tag{1}$$

where $\mathbf{x}_t^{(n)} \in \mathbb{R}^{W \times C}$ is a sensor window and $y_t^{(n)} \in [K]$ is its activity label. For each target window, the model observes up to L preceding windows from the same sequence,

$$\mathcal{H}_t^{(n)} = (\mathbf{x}_{t-L}^{(n)}, \ldots, \mathbf{x}_{t-1}^{(n)}), \qquad \mathbf{m}_t^{(n)} \in \{0,1\}^L, \tag{2}$$

where $\mathbf{m}_t^{(n)}$ marks valid history positions. We omit the sequence superscript when clear. If no valid history exists, the model reduces to local recognition.

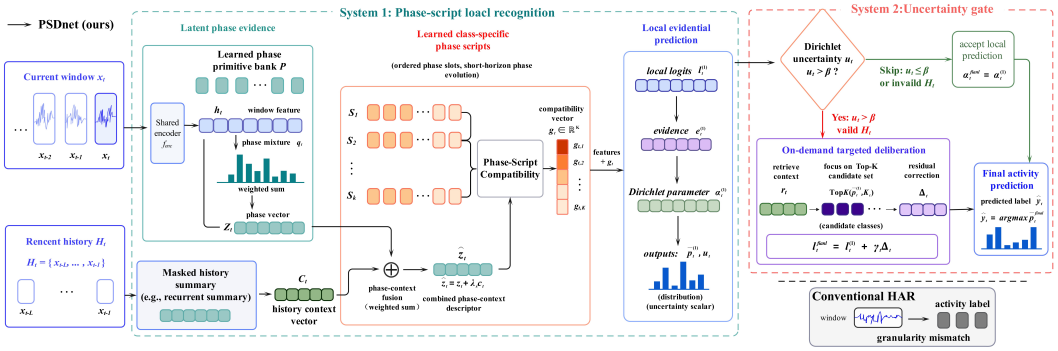


Figure 1: Overview of PSDNet. System 1 maps the current window and its short-term history into latent phase evidence, evaluates compatibility with class-specific phase scripts, and produces an evidential local prediction. The resulting uncertainty controls whether System 2 is activated. For uncertain windows, System 2 retrieves targeted context and applies a residual correction, with additional capacity focused on a small candidate set. A boundary branch provides auxiliary transition supervision but is not used as a hard routing rule.

3.2 System 1: Phase-Script Local Recognition

A shared encoder maps the current window to $\mathbf{h}_t = f_{\text{enc}}(\mathbf{x}_t) \in \mathbb{R}^d$. Rather than forcing an immediate activity decision from this feature, System 1 first represents the window as a mixture of M latent phase primitives:

$$\mathbf{q}_t = \text{softmax}(\mathbf{W}_\phi \mathbf{h}_t) \in \Delta^{M-1}, \qquad \mathbf{z}_t = \mathbf{q}_t^\top \mathbf{P} \in \mathbb{R}^d, \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{M \times d}$ is a learnable primitive bank. Valid history windows are encoded in the same way and summarized by a masked recurrent encoder,

$$\mathbf{c}_t = f_{\text{ctx}}(\mathbf{z}_{t-L}, \ldots, \mathbf{z}_{t-1}; \mathbf{m}_t) \in \mathbb{R}^d. \tag{4}$$

When the history is empty, $\mathbf{c}_t = \mathbf{0}$.

To model class-specific temporal evolution, each activity class k owns a learnable script $\mathbf{S}_k = [\mathbf{s}_{k,1}, \ldots, \mathbf{s}_{k,R}] \in \mathbb{R}^{R \times d}$, where each slot stores a latent phase pattern. We compare the current phase-context descriptor $\hat{\mathbf{z}}_t = \mathbf{z}_t + \lambda_{\text{s}} \mathbf{c}_t$ against every script by attention over its slots:

$$\omega_{t,k,r} = \text{softmax}_r\left(\frac{\hat{\mathbf{z}}_t^\top \mathbf{s}_{k,r}}{\sqrt{d}}\right), \qquad g_{t,k} = \frac{\hat{\mathbf{z}}_t^\top \sum_{r=1}^{R} \omega_{t,k,r} \mathbf{s}_{k,r}}{\sqrt{d}}. \tag{5}$$

The vector $\mathbf{g}_t = (g_{t,1}, \ldots, g_{t,K})$ scores how compatible the current local phase and recent history are with each class-specific script. System 1 then produces an evidential local prediction,

$$\begin{aligned} \boldsymbol{\ell}_t^{(1)} &= f_{\text{loc}}([\mathbf{h}_t; \mathbf{z}_t; \mathbf{c}_t]) + \mathbf{g}_t, \\ \mathbf{e}_t^{(1)} &= \text{softplus}(\boldsymbol{\ell}_t^{(1)}), \qquad \boldsymbol{\alpha}_t^{(1)} = \mathbf{e}_t^{(1)} + \mathbf{1}, \\ \bar{\mathbf{p}}_t^{(1)} &= \boldsymbol{\alpha}_t^{(1)} / \alpha_{0,t}^{(1)}. \end{aligned} \tag{6}$$

where $\alpha_{0,t}^{(1)} = \sum_k \alpha_{t,k}^{(1)}$. A lightweight boundary head $\ell_t^{\text{bnd}} = f_{\text{bnd}}(\xi_t)$ is also attached to System 1, where ξ_t contains phase-shift, phase-entropy, and prediction-margin cues defined in Appendix A.3. This branch supplies auxiliary transition supervision only and is never used as a hard routing rule.

3.3 System 2: Uncertainty-Triggered Deliberation

The local Dirichlet posterior induces a scalar ambiguity measure

$$u_t = \mathbb{E}_{\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha}_t^{(1)})}\left[-\sum_{k=1}^{K} \pi_k \log \pi_k\right], \tag{7}$$

with closed form given in Appendix A.4. Larger u_t means that the local phase evidence is less decisive. Training uses a differentiable soft gate, whereas inference evaluates System 2 only when the uncertainty exceeds the learned threshold:

$$\gamma_t = \begin{cases} \mathbb{I}[\|\mathbf{m}_t\|_1 > 0]\, \sigma(\tau(\mathrm{sg}(u_t) - \beta)), & \text{training,} \\ \mathbb{I}[\|\mathbf{m}_t\|_1 > 0]\, \mathbb{I}[u_t > \beta]\, \sigma(\tau(u_t - \beta)), & \text{inference,} \end{cases} \tag{8}$$

where β is a learnable threshold, τ is a temperature, and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The stop-gradient prevents the local predictor from lowering its own uncertainty merely to avoid deliberation.

For windows routed to System 2, a retrieval module obtains targeted context $\mathbf{r}_t = f_{\mathrm{ret}}(\mathbf{h}_t, \mathbf{h}_{t-L:t-1}, \mathbf{m}_t)$ from adapted current and historical features. We first project the window feature into the System 2 subspace: $\tilde{h}_t = f_{\text{adapt}}(h_t)$, where f_{adapt} is a single-layer MLP with layer normalization. System 2 predicts a residual correction rather than replacing the local decision:

$$\begin{aligned} \mathcal{C}_t &= \mathrm{TopK}(\bar{\mathbf{p}}_t^{(1)}, K_c), \\ \boldsymbol{\Delta}_t &= f_\Delta([\tilde{\mathbf{h}}_t; \mathbf{r}_t; \mathbf{z}_t]) + \mathrm{Scatter}_{k \in \mathcal{C}_t}(f_{\mathrm{sub}}([\mathbf{h}_t; \mathbf{z}_t; \mathbf{r}_t; \mathbf{e}_k])). \end{aligned} \tag{9}$$

Here $K_c \ll K$, $\mathbf{e}_k$ is a learnable class embedding, and $\mathrm{Scatter}$ inserts candidate-specific corrections into the full K-dimensional label space with zeros elsewhere. The dense term keeps all classes reachable, while the candidate term allocates extra capacity to likely local confusions. The final prediction is

$$\boldsymbol{\ell}_t^{\mathrm{final}} = \boldsymbol{\ell}_t^{(1)} + \gamma_t \boldsymbol{\Delta}_t, \qquad \boldsymbol{\alpha}_t^{\mathrm{final}} = \mathrm{softplus}(\boldsymbol{\ell}_t^{\mathrm{final}}) + \mathbf{1}, \qquad \bar{\mathbf{p}}_t^{\mathrm{final}} = \boldsymbol{\alpha}_t^{\mathrm{final}} / \alpha_{0,t}^{\mathrm{final}}. \tag{10}$$

If $\gamma_t = 0$, the final prediction reduces to the System 1 decision.

3.4 Training Objective

The main objective is deliberately compact. Following the evidential deep learning framework Sensoy et al. [2018], we train the Dirichlet parameters via the expected cross-entropy loss with KL regularization. For an evidential prediction $\boldsymbol{\alpha}_t$ and label y_t, we use

$$\begin{aligned} \mathcal{L}_{\mathrm{EDL}}(\boldsymbol{\alpha}_t, y_t) &= \psi(\alpha_{0,t}) - \psi(\alpha_{t,y_t}) + \eta_e \lambda_{\mathrm{KL}}\, \mathrm{KL}(\mathrm{Dir}(\tilde{\boldsymbol{\alpha}}_t) \| \mathrm{Dir}(\mathbf{1})), \\ \tilde{\boldsymbol{\alpha}}_t &= \mathbf{y}_t + (\mathbf{1} - \mathbf{y}_t) \odot \boldsymbol{\alpha}_t. \end{aligned} \tag{11}$$

where $\alpha_{0,t} = \sum_k \alpha_{t,k}$, $\mathbf{y}_t$ is the one-hot label, and $\eta_e = \min(1, e/(0.4E))$ anneals the KL term over training. The task loss supervises both the local and final predictors:

$$\mathcal{L}_{\mathrm{task}} = \frac{1}{B} \sum_{i=1}^{B} \left[\mathcal{L}_{\mathrm{EDL}}(\boldsymbol{\alpha}_i^{(1)}, y_i) + \mathcal{L}_{\mathrm{EDL}}(\boldsymbol{\alpha}_i^{\mathrm{final}}, y_i) \right]. \tag{12}$$

The first term keeps System 1 predictive and its uncertainty meaningful; the second trains the end-to-end predictor used at test time.

Two low-weight structural regularizers are added. The boundary branch is trained with class-balanced binary cross-entropy $\mathcal{L}_{\mathrm{bnd}}$ using the target $b_t = \mathbb{I}[t^- \text{ exists and } y_t \neq y_{t^-}]$, where t^- is the previous valid window in the same sequence. To preserve selectivity, we penalize only excessive routing:

$$\mathcal{L}_{\mathrm{bud}} = \left[\max(0, \bar{\gamma} - \rho) \right]^2, \qquad \bar{\gamma} = |\mathcal{V}|^{-1} \sum_{i \in \mathcal{V}} \gamma_i, \quad \mathcal{V} = \{i : \|\mathbf{m}_i\|_1 > 0\}. \tag{13}$$

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{bnd}} \mathcal{L}_{\mathrm{bnd}} + \lambda_{\mathrm{bud}} \mathcal{L}_{\mathrm{bud}}. \tag{14}$$

This formulation makes the hierarchy of the method explicit: PSDNet is defined by phase-script local recognition and uncertainty-triggered residual deliberation, with boundary and budget terms serving as light structural regularizers. Optional stabilizers for implementation are reported in Appendix A.5 rather than promoted to the main objective.

Table 1: Main benchmark results (%) on eight HAR datasets. Best results are highlighted as **first**, **second**, and **third**. Results are averaged over 10 independent runs with different random seeds.

Model \ Dataset	UCI		USC		PAMAP2		DSADS		HAPT		UniMiB		MotionSense		mHealth	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
CNN	**96.84 ± 0.21**	**96.84 ± 0.20**	93.73 ± 0.22	93.67 ± 0.24	90.49 ± 0.88	90.41 ± 0.91	84.37 ± 1.35	84.59 ± 1.38	94.55 ± 0.16	94.51 ± 0.16	75.85 ± 0.57	75.32 ± 0.57	93.87 ± 0.54	93.88 ± 0.54	98.21 ± 0.74	98.20 ± 0.76
Residual	**97.11 ± 0.29**	**97.10 ± 0.29**	93.47 ± 0.38	93.37 ± 0.38	**90.95 ± 0.55**	**90.82 ± 0.61**	85.10 ± 1.55	85.24 ± 1.53	**94.82 ± 0.18**	**94.77 ± 0.20**	76.32 ± 0.49	75.77 ± 0.57	94.73 ± 0.22	94.71 ± 0.22	97.46 ± 1.05	97.43 ± 1.07
DeepConvLSTM	96.01 ± 0.24	95.99 ± 0.24	93.55 ± 0.17	93.44 ± 0.22	88.56 ± 0.76	88.47 ± 0.73	86.42 ± 1.40	86.37 ± 1.18	94.40 ± 0.30	94.35 ± 0.29	**77.26 ± 0.32**	**76.96 ± 0.32**	95.52 ± 0.48	95.51 ± 0.49	97.57 ± 2.48	97.54 ± 2.55
Crossformer	93.99 ± 0.38	94.00 ± 0.39	92.62 ± 0.24	92.44 ± 0.24	85.78 ± 1.29	85.45 ± 1.42	83.99 ± 1.54	84.16 ± 1.60	92.80 ± 0.26	92.74 ± 0.26	72.70 ± 0.63	72.50 ± 0.57	93.93 ± 0.28	93.90 ± 0.29	92.15 ± 1.41	91.85 ± 1.49
MICN	92.90 ± 0.49	92.86 ± 0.50	90.11 ± 0.43	89.94 ± 0.48	85.58 ± 1.46	85.28 ± 1.48	82.08 ± 1.61	81.92 ± 1.72	91.68 ± 0.58	91.69 ± 0.60	70.85 ± 0.45	70.78 ± 0.40	94.06 ± 0.42	94.07 ± 0.41	95.46 ± 2.14	95.36 ± 2.23
TimeMixer	92.58 ± 0.65	92.53 ± 0.64	93.12 ± 0.24	93.03 ± 0.26	86.78 ± 2.30	86.82 ± 2.26	83.82 ± 1.98	83.57 ± 1.98	90.61 ± 0.41	90.56 ± 0.40	71.48 ± 0.37	71.12 ± 0.34	92.08 ± 0.97	92.05 ± 0.93	87.66 ± 4.52	87.28 ± 4.54
SWL-Adapt	94.93 ± 0.32	94.92 ± 0.31	**94.47 ± 0.14**	**94.34 ± 0.12**	87.08 ± 1.78	86.77 ± 1.73	86.36 ± 0.66	86.34 ± 0.55	93.56 ± 0.26	93.72 ± 0.27	74.62 ± 0.36	73.89 ± 0.33	93.21 ± 1.05	93.22 ± 1.00	94.82 ± 1.10	94.62 ± 1.16
MSGNet	92.37 ± 1.07	92.35 ± 1.06	91.19 ± 0.60	91.00 ± 0.59	84.18 ± 3.25	84.14 ± 3.11	83.59 ± 1.15	83.47 ± 1.10	91.59 ± 0.67	91.50 ± 0.66	69.73 ± 0.81	69.41 ± 0.79	92.93 ± 0.94	92.92 ± 0.96	91.75 ± 3.85	91.52 ± 4.04
CALANet	96.42 ± 0.76	96.41 ± 0.75	93.97 ± 0.35	93.82 ± 0.37	87.93 ± 0.96	87.70 ± 1.05	87.53 ± 0.68	87.47 ± 0.65	93.67 ± 0.70	93.74 ± 0.67	**78.66 ± 0.60**	**78.14 ± 0.63**	**95.76 ± 0.58**	**95.76 ± 0.58**	**98.96 ± 0.34**	**98.96 ± 0.34**
RelCon	94.83 ± 0.40	94.81 ± 0.40	94.41 ± 0.22	94.28 ± 0.26	89.68 ± 1.04	89.67 ± 1.02	**87.81 ± 1.17**	**87.65 ± 1.14**	93.69 ± 0.34	93.76 ± 0.32	76.89 ± 0.62	76.50 ± 0.60	92.54 ± 0.99	92.58 ± 0.99	96.12 ± 2.03	96.01 ± 2.14
TimeFilter	87.72 ± 0.77	87.73 ± 0.78	71.61 ± 1.40	71.49 ± 1.38	80.62 ± 0.75	80.44 ± 0.81	82.00 ± 0.80	81.55 ± 0.84	87.78 ± 0.78	87.65 ± 0.77	67.39 ± 0.68	67.05 ± 0.70	91.05 ± 0.87	91.02 ± 0.88	82.77 ± 0.84	82.56 ± 0.90
DeepConvContext	96.60 ± 0.65	96.58 ± 0.65	**94.88 ± 0.36**	**94.80 ± 0.36**	**94.99 ± 0.87**	**94.94 ± 0.89**	**87.85 ± 1.18**	87.54 ± 1.13	**96.69 ± 0.38**	**96.69 ± 0.37**	73.67 ± 1.03	73.35 ± 0.96	**96.39 ± 0.60**	**96.39 ± 0.61**	98.87 ± 0.35	98.87 ± 0.35
MoPFormer	93.01 ± 0.36	93.00 ± 0.34	93.68 ± 0.22	93.56 ± 0.24	85.96 ± 1.70	85.91 ± 1.57	84.74 ± 0.90	84.77 ± 0.86	91.60 ± 0.35	91.63 ± 0.35	70.91 ± 1.09	70.61 ± 1.07	94.12 ± 0.57	94.13 ± 0.56	97.51 ± 1.96	97.47 ± 2.05
PSDNet	**97.84 ± 0.22**	**97.83 ± 0.22**	**96.66 ± 0.26**	**96.63 ± 0.27**	**96.55 ± 0.44**	**96.52 ± 0.46**	**89.57 ± 0.68**	**89.17 ± 0.76**	**97.39 ± 0.16**	**97.35 ± 0.20**	**80.78 ± 0.81**	**80.37 ± 0.79**	**97.04 ± 0.42**	**97.03 ± 0.42**	**99.04 ± 0.06**	**99.04 ± 0.06**

4 Experiments

4.1 Experimental Setup

We evaluate PSDNet on a diverse set of public HAR benchmarks that cover both relatively clean daily-activity datasets and more challenging settings with stronger inter-class similarity, longer-range temporal structure, or higher transition ambiguity. The benchmarks considered in this study include UCI Anguita et al. [2013], USC Zhang and Sawchuk [2012b], PAMAP2 Reiss and Stricker [2012], DSADS Altun et al. [2010], HAPT Reyes-Ortiz et al. [2016], UniMiB Micucci et al. [2017], MotionSense Malekzadeh et al. [2019], and mHealth Banos et al. [2015]. Together, these datasets span a broad range of activity granularity, sensor configurations, and recognition difficulty.

We compare against representative baselines from two families: CNN-based HAR models (standard CNN Huang et al. [2021], residual CNN variant Cheng et al. [2022], and DeepConvLSTM Ordóñez and Roggen [2016]) and recent sequence models (Crossformer Zhang and Yan [2023], MICN Wang et al. [2023], TimeMixer Wang et al. [2024a], SWL-Adapt Hu et al. [2023], MSGNet Cai et al. [2024], CALANet Park et al. [2024], RelCon Xu et al. [2024], TimeFilter Hu et al. [2025], DeepConvContext Bock et al. [2025], and MoPFormer Zhang et al. [2025]). All methods are evaluated with standard classification metrics, and Table 1 reports accuracy and weighted F1. Reported values are means with standard deviations across all completed runs.

4.2 Main Benchmark Results

Table 1 summarizes the main cross-dataset comparison. PSDNet achieves the best results across all benchmark datasets in both accuracy and weighted F1. The gains are not confined to one benchmark type: they appear on relatively clean datasets such as UCI and mHealth, but they are more pronounced on harder settings such as USC, PAMAP2, HAPT, UniMiB, and MotionSense, where direct window-level classification is more likely to suffer from partial observations and look-alike local patterns.

This pattern is consistent with the central thesis of the paper. When the window already contains sufficiently distinctive evidence, PSDNet remains competitive without needing unnecessary extra reasoning. When the local window is incomplete or ambiguous, however, phase-level interpretation and selective script-aware deliberation provide a clearer advantage. In particular, the strongest improvements over the best non-PSDNet baseline are substantial on PAMAP2, UniMiB, and MotionSense, suggesting that the proposed formulation is especially beneficial when local motion fragments are insufficient for reliable direct classification. Statistical significance tests are reported in Appendix B.2

4.3 Ablation Study

Table 2 ablates the main components of PSDNet on UniMiB and PAMAP2. All component removals reduce accuracy and weighted F1, confirming that PSDNet benefits from the combined design of temporal context, phase modeling, and selective deliberation. Removing history context gives the largest drop on both datasets, showing that recent window history is essential for interpreting partial local evidence. Removing phase primitives also substantially degrades performance, especially on PAMAP2, and strongly affects routing on UniMiB, where the trigger rate falls from 42.2% to 0.1%. System 2 contributes more on UniMiB than on PAMAP2, consistent with the higher trigger rate and the larger fraction of windows requiring deliberation. Class scripts, dense correction, and boundary supervision provide smaller but consistent gains, indicating that phase-script compatibility and auxiliary transition sensitivity further refine the classification. Overall, these results show that

Table 2: Ablation study of different model variants on two datasets. (Results are averaged over 5 independent runs with different random seeds.)

Variants \ Dataset	UniMiB				PAMAP2			
	Acc(%)	**F1(%)**	**Trigger Rate(%)**	**Latency (ms)**	**Acc(%)**	**F1(%)**	**Trigger Rate(%)**	**Latency (ms)**
w/o Phase Primitives	78.87 ± 0.54	78.54 ± 0.60	0.1	0.091	94.84 ± 0.52	94.78 ± 0.54	2.1	0.491
w/o Class Scripts	80.45 ± 0.90	79.89 ± 0.97	40.7	0.125	96.83 ± 0.34	96.81 ± 0.34	0.6	0.472
w/o History Context	75.68 ± 0.43	75.47 ± 0.33	0.0	0.059	93.88 ± 0.33	93.79 ± 0.33	0.0	0.137
w/o System 2	78.19 ± 0.42	77.84 ± 0.41	0.0	0.086	96.47 ± 0.55	96.43 ± 0.56	0.0	0.475
w/o Dense Correction	79.39 ± 0.65	78.96 ± 0.75	25.4	0.146	96.59 ± 0.40	96.56 ± 0.44	0.4	0.477
w/o Boundary Loss	79.43 ± 0.60	79.12 ± 0.60	39.9	0.143	96.72 ± 0.28	96.68 ± 0.28	0.4	0.474
PSDNet	80.50 ± 0.78	80.14 ± 0.55	42.2	0.146	97.08 ± 0.39	97.07 ± 0.39	1.3	0.488

Table 3: Ablation Study of Triggering Strategies on Two Datasets. (Results are averaged over 5 independent runs with different random seeds.)

Variants \ Dataset	UniMiB				PAMAP2			
	Acc(%)	**F1(%)**	**Trigger Rate(%)**	**Latency (ms)**	**Acc(%)**	**F1(%)**	**Trigger Rate(%)**	**Latency (ms)**
w/o System 2	78.19 ± 0.42	77.84 ± 0.41	0.0	0.086	96.47 ± 0.55	96.43 ± 0.56	0.0	0.475
Always System 2	80.51 ± 1.39	80.10 ± 1.49	100.0	0.157	97.45 ± 0.55	97.44 ± 0.57	100.0	0.508
Random Gate	80.13 ± 0.59	79.69 ± 0.43	44.5	0.158	96.74 ± 0.68	96.70 ± 0.70	1.0	0.470
CE+Margin Gate	80.32 ± 0.43	79.81 ± 0.30	42.2	0.212	96.71 ± 0.18	96.70 ± 0.18	1.5	0.471
PSDNet	80.50 ± 0.78	80.14 ± 0.55	42.2	0.146	97.08 ± 0.39	97.07 ± 0.39	1.3	0.488

the gains of PSDNet arise from the intended mechanism: phase-aware local evidence construction followed by uncertainty-gated deliberation for difficult windows.

Table 3 examines whether the gain from System 2 stems from targeting the right windows rather than applying more computation overall. On PAMAP2, Always System 2 gives the numerically highest accuracy and F1 (97.45%/97.44%), but it deliberates on every window and incurs the highest latency (0.508 ms). In contrast, PSDNet reaches 97.08% accuracy and 97.07% F1 while triggering on only 1.3% of windows, retaining most of the full-deliberation performance at a much smaller computational budget. Random gating at a matched trigger budget (1.0%) trails PSDNet by 0.34 points in accuracy. Replacing the evidential gate with softmax-margin thresholding at a comparable rate (1.5%) also underperforms (96.71% vs. 97.08%), suggesting that Dirichlet-based uncertainty better identifies genuinely incomplete local evidence. On UniMiB, PSDNet matches the full-deliberation accuracy within 0.01 points and slightly improves F1, while using 42.2% rather than 100% of the System 2 calls. The learned gate therefore provides a better accuracy–efficiency trade-off, and automatically adapts its trigger rate to dataset difficulty (42.2% on UniMiB vs. 1.3% on PAMAP2) without manual tuning.

4.4 In-Depth Empirical Analysis

Boundary-Stratified Evaluation. We stratify test samples into boundary, near-boundary, and interior groups by their distance to ground-truth label transitions. As shown in Table 4, PSDNet improves overall F1 on both datasets, with particularly clear gains in transition-adjacent regions. On PAMAP2, boundary F1 rises from 45.70% to 62.40%, and near-boundary F1 rises from 74.50% to 88.90%, indicating a substantial improvement near activity transitions. On HAPT, boundary F1 improves by 3.38 points, and the gains are consistent across boundary, near-boundary, and interior samples. These results suggest that PSDNet improves recognition performance across different regions of the sequence, with the strongest evidence for enhanced transition modeling coming from the boundary and near-boundary gains on PAMAP2 and HAPT.

Sensitivity to Window Construction. On MotionSense, we compare PSDNet with a matched-history HistoryMHA control under native, shorter, and non-overlapping windows while holding the physical history budget fixed at 7.68 seconds. PSDNet achieves 97.15%, 97.12%, and 96.58% weighted F1, outperforming HistoryMHA by 0.39, 0.73, and 0.81 points, respectively. When the test segmentation grid is shifted by up to 0.75 times the native stride without retraining, PSDNet continues to outperform HistoryMHA by 0.30–0.39 points. Complete settings and statistical tests are provided in Appendix B.7.

Table 4: Weighted F1 (%) stratified by proximity to activity transition boundaries.

Dataset	Method	Boundary F1	Near-boundary F1	Interior F1	Overall F1	#Bound	#Near	#Interior
HAPT	Baseline	90.32 ± 1.02	96.48 ± 0.24	97.88 ± 0.24	95.92 ± 0.21	605	955	1596
	PSDNet	93.70 ± 0.81	97.56 ± 0.54	98.60 ± 0.46	97.32 ± 0.46	605	955	1596
PAMAP2	Baseline	45.70 ± 3.20	74.50 ± 2.10	95.60 ± 0.40	93.20 ± 0.30	52	104	1892
	PSDNet	62.40 ± 5.60	88.90 ± 3.70	97.70 ± 0.40	96.30 ± 0.60	52	104	1892

System 2 Trigger Quality Diagnostics. Fig.2 illustrates the deliberation timeline of PSDNet on representative test sequences from (a) HAPT and (b) PAMAP2. System 1 maintains low entropy throughout stable activity segments, while uncertainty spikes arise around transition-adjacent or locally ambiguous windows. The threshold β activates System 2 only sparsely in these representative timelines; aggregate trigger-rate statistics are reported in Table 3 and Appendix B.3. As the Pred Status panels confirm, System 2 corrects mispredictions at critical junctures (teal bars) with negligible over-correction (orange bars), demonstrating an effective accuracy–efficiency trade-off in the uncertainty-gated dual-system design. Additional quantitative analysis of the gate's selectivity is provided in Appendix B.3.

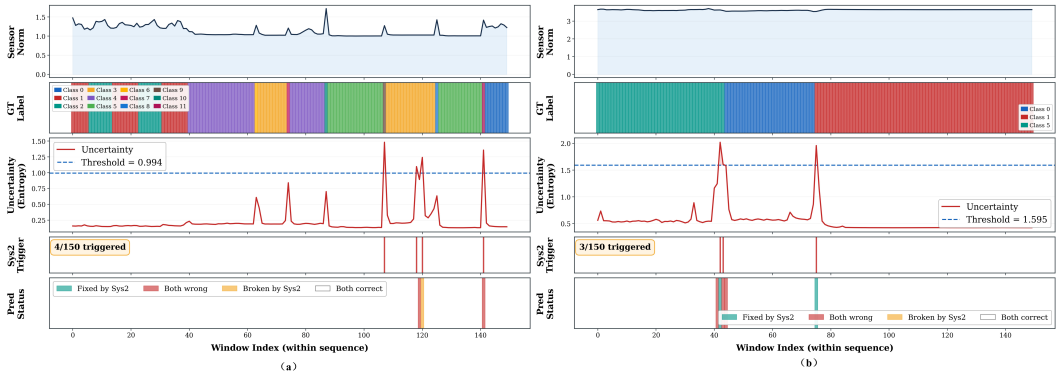


Figure 2: System 2 Deliberation Timelines on HAPT and PAMAP2. Temporal visualization of the PSDNet deliberation pipeline on representative test sequences from (a) HAPT and (b) PAMAP2. All five panels share a common horizontal axis denoting the sliding window index t within the test sequence. From top to bottom, the vertical axes represent: Sensor Norm, the L_2 norm of the sensor feature vector; GT Label, color-coded ground-truth activity class; Uncertainty, Dirichlet-based predictive entropy from System 1, with the dashed line denoting the threshold β; Sys2 Trigger, windows where System 2 deliberation is activated; Pred Status, per-window outcome (Fixed by Sys2: corrected; Both wrong: both systems incorrect; Broken by Sys2: overridden incorrectly; Both correct).

Look-Alike Class Confusion Analysis. To examine whether PSDNet genuinely improves disambiguation of semantically similar activities, we automatically extract the top-3 most confused class pairs from the baseline confusion matrix for each dataset. As reported in Table 5, PSDNet reduces the mutual confusion rate on all 9 pairs across three datasets. The largest absolute reduction appears on USC (class 1 vs. 2, 7.91%) and UniMiB (class 9 vs. 13, 6.99%), with class 2 vs. 11 on USC being completely resolved. These results confirm that the phase-aware class scripts capture discriminative temporal patterns essential for resolving locally similar but globally distinct activities.

Table 5: Look-alike Class Pair Confusion Analysis.

Dataset	Confusing pair	Baseline(%)	PSDNet(%)	Reduction(%)
PAMAP2	3 vs 11	7.33	3.41	3.92
	2 vs 3	5.03	1.32	3.70
	1 vs 2	3.59	1.49	2.10
UniMiB	9 vs 13	22.40	15.41	6.99
	11 vs 14	20.60	17.31	3.28
	10 vs 15	18.32	14.64	3.68
USC	1 vs 2	35.80	27.89	7.91
	1 vs 11	2.88	0.21	2.66
	2 vs 11	1.18	0.00	1.18

Accuracy–Efficiency Trade-off. Fig. 3 presents the Pareto frontier of PSDNet on PAMAP2 under three efficiency metrics, including GFLOPs, inference latency, and System 2 trigger rate. By varying the uncertainty threshold, PSDNet traces a curve ranging from System 1 only to full deliberation. The F1 score remains nearly constant across a wide range of operating points. A trigger rate below 10% already achieves an F1 score within 0.1% of the peak value (97.1%), while using fewer GFLOPs than full deliberation. This result demonstrates that PSDNet attains near-optimal accuracy with substantially lower computational cost through selective deliberation. A more detailed analysis of the accuracy–efficiency trade-off on the HAPT and USC datasets is presented in the Appendix B.4.

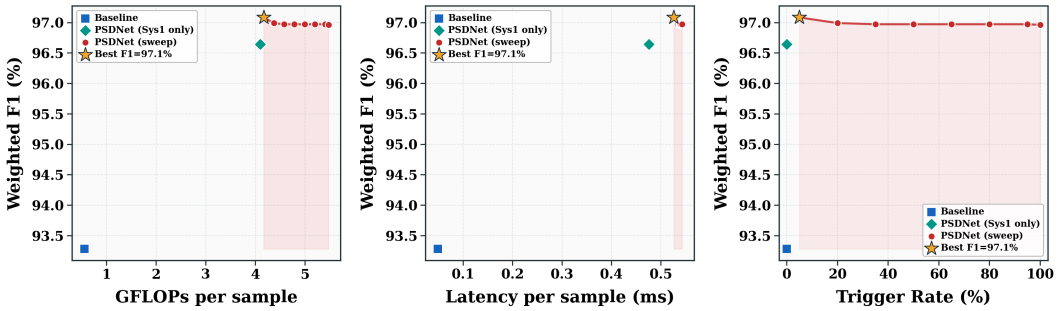


Figure 3: Accuracy–efficiency Pareto analysis of PSDNet under varying System 2 trigger rates on PAMAP2. Left: Weighted F1 vs. GFLOPs per sample, middle: Weighted F1 vs. inference latency per sample, right: Weighted F1 vs. System 2 trigger rate. Blue squares: CNN baseline, teal diamonds: PSDNet (System 1 only), red circles: PSDNet with varying uncertainty thresholds, stars: best operating point. Shaded regions indicate the Pareto-dominant area.

Phase-Script Visualization. Fig.4 presents the internal phase-script dynamics of PSDNet on two representative test sequences. The primitive-mixture activation heatmap (row 3) illustrates how local phase evidence varies across stable activity segments and around transitions. The uncertainty curve (row 4) remains low within intra-activity regions and spikes near activity boundaries, with only a few windows exceeding the adaptive threshold to trigger the System-2 refinement pathway. This confirms that the dual-system mechanism activates selectively at genuinely ambiguous transitions, preserving computational efficiency while improving classification accuracy at critical decision points. Further analysis of the phase activation patterns and their correlation with activity transitions is presented in the Appendix B.5.

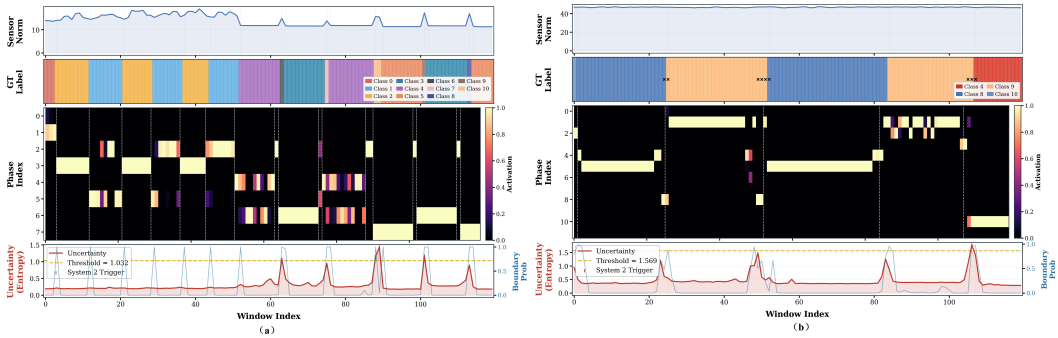


Figure 4: Phase-script dynamics visualization on (a) HAPT and (b) PAMAP2. The horizontal axis denotes the window index in the test sequence. Each panel shows four rows: (1) Sensor Norm, the L2-norm of raw sensor readings, (2) GT Label, ground-truth activity classes as color-coded bands, with misclassified windows highlighted, (3) Phase Activation Heatmap, the learned phase probability distribution (magma colormap), and (4) Uncertainty & Boundary, entropy-based uncertainty (red, left axis) and boundary transition probability (blue, right axis), with the adaptive threshold as a yellow dashed line. Vertical dashed lines mark ground-truth transition boundaries.

The correct class ranks first in the script branch substantially more often than chance on both datasets (Table 6). Primitive evidence changes more sharply near transitions than in interior regions, and class compatibility shifts toward the incoming activity. These measurements complement the qualitative examples in Fig. 4. Chronological-history perturbation is reported in Appendix B.5.

Table 6: Phase-Script Diagnostics. Script Top-1 is the proportion of windows whose correct class has the highest compatibility score (chance: 8.33%). Primitive JS B/I and N/I compare adjacent-window divergence at boundaries and near-boundaries with interior regions. Positive shifts favor the incoming activity.

Dataset	Script Top-1 (%)	Primitive JS B/I	Primitive JS N/I	Transition compatibility shift
HAPT	84.82 ± 12.74	$19.45\times$	$1.86\times$	$+6.96 \pm 2.78$
PAMAP2	57.75 ± 20.53	$5.51\times$	$6.14\times$	$+1.14 \pm 1.01$

5 Discussion and Limitations

PSDNet is motivated by a simple abstraction shift: a sliding window should be viewed as partial evidence within an evolving activity rather than as a complete action instance. The empirical results support this view across main benchmarks, module ablations, trigger diagnostics, look-alike class analysis, and accuracy–efficiency trade-offs. In particular, the gains are most informative when local evidence is incomplete, transition-adjacent, or shared by multiple activity classes, suggesting that phase-script compatibility and selective deliberation provide a useful alternative to direct window-level classification.

The analysis also clarifies the operating regime in which PSDNet is most beneficial. Rather than applying heavier temporal reasoning uniformly to all windows, PSDNet concentrates additional computation on windows where local phase evidence is uncertain and class-specific temporal evolution is informative. This selective design opens several directions for future work, including richer phase-script structures, more interpretable phase primitives, and adaptive deliberation policies for longer streaming sequences. More broadly, PSDNet suggests that sliding-window HAR can move beyond direct window-to-label mapping toward recognition systems that reason over partial evidence, temporal compatibility, and computational selectivity jointly.

6 Acknowledgments and Disclosure of Funding

This work was supported by the Special Research Project for Building a Strong Education Province in Henan, China, entitled "A Study on the Classification-Based Evaluation of the Contributions of Philosophy and Social Science Research Outcomes to Their Translation and Application at Henan Universities and on Data Governance Mechanisms."

References

Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C Ranasinghe. Attend and discriminate: Beyond the state-of-the-art for human activity recognition using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–22, 2021.

Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pages 3–4, 2013.

Oresti Banos, Claudia Villalonga, Rafael Garcia, Alejandro Saez, Miguel Damas, Juan A Holgado-Terriza, Sungyong Lee, Hector Pomares, and Ignacio Rojas. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online*, 14(Suppl 2):S6, 2015.

Carlos Betancourt, Wen-Hui Chen, and Chi-Wei Kuan. Self-attention networks for human activity recognition using wearable devices. In *2020 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 1194–1199. IEEE, 2020.

Marius Bock, Michael Moeller, and Kristof Van Laerhoven. Temporal action localization for inertial-based human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4):1–19, 2024.

Marius Bock, Michael Moeller, and Kristof Van Laerhoven. Deepconvcontext: A multi-scale approach to timeseries classification in human activity recognition. *arXiv preprint arXiv:2505.20894*, 2025.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11141–11149, 2024.

Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021.

Lei Chen, Shurui Fan, Vikram Kumar, and Yating Jia. A method of human activity recognition in transitional period. *Information*, 11(9):416, 2020.

Xin Cheng, Lei Zhang, Yin Tang, Yue Liu, Hao Wu, and Jun He. Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices. *IEEE Sensors Journal*, 22(6):5889–5901, 2022.

Nguyen Ngoc Diep, Cuong Pham, and Tu Minh Phuong. Motion primitive forests for human activity recognition using wearable sensors. In *Pacific Rim International Conference on Artificial Intelligence*, pages 340–353. Springer, 2016.

Iveta Dirgová Luptáková, Martin Kubovčík, and Jiří Pospíchal. Wearable sensor-based human activity recognition with transformer model. *Sensors*, 22(5):1911, 2022.

Daniel Geissler, Dominique Nshimyimana, Vitor Fortes Rey, Sungho Suh, Bo Zhou, and Paul Lukowicz. Beyond confusion: A fine-grained dialectical examination of human activity recognition benchmark datasets. *arXiv preprint arXiv:2412.09037*, 2024.

Rong Hu, Ling Chen, Shenghuan Miao, and Xing Tang. Swl-adapt: An unsupervised domain adaptation model with sample weight learning for cross-user wearable human activity recognition. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 37, pages 6012–6020, 2023.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. *arXiv preprint arXiv:2501.13041*, 2025.

Wenbo Huang, Lei Zhang, Wenbin Gao, Fuhong Min, and Jun He. Shallow convolutional neural networks for human activity recognition using wearable sensors. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.

Wenbo Huang, Lei Zhang, Shuoyuan Wang, Hao Wu, and Aiguo Song. Deep ensemble learning for human activity recognition using wearable sensors via filter activation. *ACM Transactions on Embedded Computing Systems*, 22(1):1–23, 2022.

Zepeng Huo, Arash PakBin, Xiaohan Chen, Nathan Hurley, Ye Yuan, Xiaoning Qian, Zhangyang Wang, Shuai Huang, and Bobak Mortazavi. Uncertainty quantification for deep context-aware mobile activity recognition and unknown context discovery. In *International Conference on Artificial Intelligence and Statistics*, pages 3894–3904. PMLR, 2020.

Yoo-Kyung Lee, Chang-Sik Son, and Won-Seok Kang. Depthwise-separable u-net for wearable sensor-based human activity recognition. *Applied Sciences*, 15(16):9134, 2025.

Shuangjian Li, Tao Zhu, Mingxing Nie, Huansheng Ning, Zhenyu Liu, and Liming Chen. P2lhap: wearable-sensor-based human activity recognition, segmentation, and forecast through patch-to-label seq2seq transformer. *IEEE Internet of Things Journal*, 12(6):6818–6830, 2024.

Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*, pages 49–58, 2019.

Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

Jaegyun Park, Dae-Won Kim, and Jaesung Lee. Calanet: Cheap all-layer aggregation for human activity recognition. *Advances in Neural Information Processing Systems*, 37:69419–69444, 2024.

Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th international conference on pervasive technologies related to assistive environments*, pages 1–8, 2012.

Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Shuai Shao, Yu Guan, and Victor Sanchez. Beyond isolated frames: Enhancing sensor-based human activity recognition through intra-and inter-frame attention. *arXiv preprint arXiv:2405.19349*, 2024.

Yin Tang, Lei Zhang, Fuhong Min, and Jun He. Multiscale deep feature learning for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Electronics*, 70(2):2106–2116, 2022.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 119:3–11, 2019.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024a.

Zhiqiong Wang, Hanyu Liu, Boyang Zhao, Qi Shen, Mingzhe Li, Ningfeng Que, Mingke Yan, and Junchang Xin. Multi-scale context-aware networks based on fragment association for human activity recognition. In *IJCAI*, pages 3169–3177, 2024b.

Songpengcheng Xia, Lei Chu, Ling Pei, Jiarui Yang, Wenxian Yu, and Robert C Qiu. Timestamp-supervised wearable-based activity segmentation and recognition with contrastive learning and order-preserving optimal transport. *IEEE Transactions on Mobile Computing*, 23(12):10734–10751, 2024.

Maxwell A Xu, Jaya Narain, Gregory Darnell, Haraldur Hallgrimsson, Hyewon Jeong, Darren Forde, Richard Fineman, Karthik J Raghuram, James M Rehg, and Shirley Ren. Relcon: Relative contrastive learning for a motion foundation model for wearable data. *arXiv preprint arXiv:2411.18822*, 2024.

Hao Zhang, Zhan Zhuang, Xuehao Wang, Xiaodong Yang, and Yu Zhang. Mopformer: Motion-primitive transformer for wearable-sensor activity recognition. *arXiv preprint arXiv:2505.20744*, 2025.

Mi Zhang and Alexander A Sawchuk. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT international health informatics symposium*, pages 631–640, 2012a.

Mi Zhang and Alexander A Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 1036–1043, 2012b.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Yexu Zhou, Haibin Zhao, Yiran Huang, Till Riedel, Michael Hefenbrock, and Michael Beigl. Tinyhar: A lightweight deep learning model designed for human activity recognition. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers*, pages 89–93, 2022.

A Method Appendix

A.1 Notation Summary

Symbol	Meaning
N, T_n	Number of sequences and number of windows in sequence n.
W, C, K	Window length, number of sensor channels, and number of activity classes.
L, M, R, d	History length, number of phase primitives, script length, and hidden dimension.
$\mathbf{x}_t^{(n)}, y_t^{(n)}$	Window and activity label at position t of sequence n.
$\mathcal{H}_t^{(n)}, \mathbf{m}_t^{(n)}$	Historical windows and their validity mask.
$\mathbf{h}_t$	Encoder feature of the current window.
$\mathbf{q}_t, \mathbf{z}_t$	Latent phase mixture and corresponding phase feature.
$\mathbf{c}_t$	Masked summary of recent phase history.
$\mathbf{S}_k, \mathbf{g}_t$	Class-k phase script and class-wise script-compatibility vector.
$\boldsymbol{\ell}_t^{(1)}, \boldsymbol{\ell}_t^{\text{final}}$	Local logits and final corrected logits.
$\boldsymbol{\alpha}_t^{(1)}, \boldsymbol{\alpha}_t^{\text{final}}$	Local and final Dirichlet parameters.
$u_t, \beta, \tau, \gamma_t$	Local uncertainty, learnable threshold, gate temperature, and selective gate.
$\text{sg}(\cdot)$	Stop-gradient operator.
$\mathbf{r}_t$	Targeted contextual evidence retrieved by System 2.
$K_c, \mathcal{C}_t, \mathbf{e}_k$	Candidate-set size, candidate set, and candidate class embedding.
$\boldsymbol{\Delta}_t$	System 2 residual correction to the local logits.
b_t, ℓ_t^{bnd}	Auxiliary boundary target and boundary logit.
$\bar{\gamma}, \rho$	Average valid-history gate and target trigger rate.
λ's	Fusion and regularization coefficients.

Table 7: Notation summary for Section 3.

A.2 Pseudo-code for Training and Inference

A.3 Boundary Branch and Deliberation Details

The boundary head uses lightweight cues derived from local phase evidence, history, and prediction margin. Let

$$\bar{\mathbf{q}}_t^{\text{ctx}} = \sum_{i=1}^{L} \tilde{m}_{t,i} \mathbf{q}_{t-i}, \qquad \tilde{m}_{t,i} = \frac{m_{t,i}}{\sum_{j=1}^{L} m_{t,j} + \varepsilon}. \tag{15}$$

We define

$$\begin{aligned} d_t^{\text{phase}} &= \frac{1}{M} \|\mathbf{q}_t - \bar{\mathbf{q}}_t^{\text{ctx}}\|_1, \\ h_t^{\text{phase}} &= -\sum_{m=1}^{M} q_{t,m} \log q_{t,m}, \\ m_t^{\text{cls}} &= \bar{p}_{t,(1)}^{(1)} - \bar{p}_{t,(2)}^{(1)}. \end{aligned} \tag{16}$$

where $\bar{p}_{t,(1)}^{(1)}$ and $\bar{p}_{t,(2)}^{(1)}$ are the largest and second-largest entries of $\bar{\mathbf{p}}_t^{(1)}$. The boundary logit is

$$\ell_t^{\text{bnd}} = f_{\text{bnd}}([\mathbf{z}_t; \mathbf{c}_t; \mathbf{q}_t; d_t^{\text{phase}}; h_t^{\text{phase}}; m_t^{\text{cls}}]). \tag{17}$$

Input: current window $\mathbf{x}_t$, history $\mathcal{H}_t$, mask $\mathbf{m}_t$, training flag.
Output: final evidential prediction $\boldsymbol{\alpha}_t^{\text{final}}$ and auxiliary outputs.

1. Phase-script local recognition.

1. Encode $\mathbf{x}_t$ as $\mathbf{h}_t$; infer phase mixture $\mathbf{q}_t$ and phase feature $\mathbf{z}_t$.
2. Encode valid history windows and summarize them as $\mathbf{c}_t$.
3. Compare $\hat{\mathbf{z}}_t = \mathbf{z}_t + \lambda_{\text{s}}\mathbf{c}_t$ with each class script $\mathbf{S}_k$ to obtain $\mathbf{g}_t$.
4. Produce local logits $\boldsymbol{\ell}_t^{(1)}$ and local Dirichlet parameters $\boldsymbol{\alpha}_t^{(1)}$.

2. Selective routing.

1. Compute uncertainty u_t from the local Dirichlet posterior.
2. During training, use the soft gate in Eq. (8). During inference, evaluate System 2 only when $u_t > \beta$ and valid history exists.

3. Targeted deliberation.

1. If $\gamma_t = 0$, set $\boldsymbol{\Delta}_t = \mathbf{0}$.
2. Otherwise, retrieve targeted context $\mathbf{r}_t$, form the candidate set $\mathcal{C}_t$, and compute the residual correction $\boldsymbol{\Delta}_t$.
3. Return the final Dirichlet parameters from Eq. (10).

Figure 5: Training and inference procedure of PSDNet.

For minibatch boundary counts N_+ and N_-, we use $w^+ = N_-/\max(N_+, 1)$ and

$$\mathcal{L}_{\text{bnd}} = -\frac{1}{B}\sum_{i=1}^{B}\left[w^+ b_i \log\sigma(\ell_i^{\text{bnd}}) + (1-b_i)\log(1-\sigma(\ell_i^{\text{bnd}}))\right]. \tag{18}$$

For System 2, the compact retrieval function in Section 3.3 is implemented as

$$\tilde{\mathbf{h}}_t = f_{\text{adp}}(\mathbf{h}_t), \qquad \tilde{\mathbf{H}}_t = [f_{\text{adp}}(\mathbf{h}_{t-L}), \ldots, f_{\text{adp}}(\mathbf{h}_{t-1})], \tag{19}$$

$$\mathbf{G}_t = \text{GRU}(\tilde{\mathbf{H}}_t; \mathbf{m}_t), \qquad \mathbf{r}_t = \text{MHA}(\tilde{\mathbf{h}}_t, \mathbf{G}_t, \mathbf{G}_t). \tag{20}$$

The dense correction $f_\Delta([\tilde{\mathbf{h}}_t; \mathbf{r}_t; \mathbf{z}_t])$ is defined over all classes. Candidate-specific corrections are computed only for $k \in \mathcal{C}_t$ and then scattered back to the full label space. This preserves all-class reachability while concentrating extra capacity on likely confusions.

A.4 Closed-Form Details for Uncertainty and EDL

The expected categorical entropy in Eq. (7) is

$$u_t = \sum_{k=1}^{K}\frac{\alpha_{t,k}^{(1)}}{\alpha_{0,t}^{(1)}}\left[\psi(\alpha_{0,t}^{(1)}+1) - \psi(\alpha_{t,k}^{(1)}+1)\right]. \tag{21}$$

The Dirichlet-to-uniform KL term in Eq. (11) is

$$\begin{aligned}\text{KL}(\text{Dir}(\boldsymbol{\alpha})\|\text{Dir}(\mathbf{1})) = \log\Gamma(\alpha_0) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) - \log\Gamma(K) \\ + \sum_{k=1}^{K}(\alpha_k - 1)\left(\psi(\alpha_k) - \psi(\alpha_0)\right).\end{aligned} \tag{22}$$

where $\alpha_0 = \sum_k \alpha_k$.

A.5 Optional Training Stabilizers

The following terms are optional implementation stabilizers rather than core components of Section 3. To reduce early under-training of System 2, one may supervise the ungated corrective branch through

$$\boldsymbol{\ell}_t^{\text{aux}} = \text{sg}(\boldsymbol{\ell}_t^{(1)}) + \boldsymbol{\Delta}_t, \qquad \mathcal{L}_{\text{aux}} = \frac{1}{B}\sum_{i=1}^{B}\mathcal{L}_{\text{EDL}}(\text{softplus}(\boldsymbol{\ell}_i^{\text{aux}}) + \mathbf{1}, y_i). \tag{23}$$

Table 8: Dataset preprocessing and held-out evaluation protocols. The windowing settings and split units follow commonly used configurations in the cited sensor-based HAR studies and dataset descriptions. Channels denote the retained input channels after preprocessing, while window size and overlap rate specify the sliding-window construction of model inputs.

Dataset	#Subjects	#Classes	Channels	Sampling Rate	Window Size	Overlap Rate	Split Unit	Train:Test
UCI Huang et al. [2022]	30	6	9	50 Hz	128	50%	Subject	7:3
HAPT Reyes-Ortiz et al. [2016]	30	12	6	50 Hz	128	50%	Subject	7:3
PAMAP2 Tang et al. [2022]	9	12	40	100 Hz	171	78%	Subject	4:1
UniMiB Micucci et al. [2017]	30	17	3	50 Hz	151	50%	Subject	7:3
USC Zhang and Sawchuk [2012b]	14	12	6	100 Hz	512	50%	Subject	4:1
MotionSense Malekzadeh et al. [2019]	24	6	6	50 Hz	128	50%	Subject	4:1
DSADS Altun et al. [2010]	8	19	45	25 Hz	125	50%	Subject	3:1
mHealth Lee et al. [2025]	10	12	12	50 Hz	128	50%	Subject	4:1

To discourage collapse of phase usage, one may regularize the minibatch-average phase mixture,

$$\bar{\mathbf{q}} = \frac{1}{B}\sum_{i=1}^{B}\mathbf{q}_i, \qquad \mathcal{L}_{\text{div}} = \mathrm{KL}\left(\bar{\mathbf{q}} \| \frac{1}{M}\mathbf{1}\right). \tag{24}$$

B Experimental Appendix

B.1 Experimental Details

Experimental Settings. Our experiments are implemented in PyTorch and conducted on a workstation equipped with an NVIDIA A40 GPU. Model optimization is performed using AdamW with an initial learning rate of 1×10^{-3}, a weight decay of 1×10^{-4}, and a cosine annealing learning-rate schedule. We set the batch size to 64 and train each model for at most 120 epochs. The hidden dimension d is fixed to 256, while the script length R is set to 4. For the dataset-specific hyperparameters, we set the number of phase primitives M, the history context length L, and the gate temperature τ according to the configuration of each dataset. Main benchmark results are averaged over 10 independent seeds.

Dataset preprocessing and evaluation protocol. For each dataset, we adopt the preprocessing configuration and evaluation unit commonly used in prior sensor-based HAR studies and dataset-specific protocols, as summarized in Table 8. Raw sensor recordings are first organized according to their original subject or session identifiers. Sliding-window segmentation is then performed independently within each original recording, using the dataset-specific window size and overlap rate reported in Table 8. Windows are never generated across subject or session boundaries.

Each generated window retains the subject/session identifier of its source recording. The held-out train/test partition is constructed at the subject/session level: all windows derived from the same subject or session are assigned exclusively to either the training partition or the held-out test partition. Therefore, overlapping neighboring windows from the same original recording cannot appear in both training and test partitions.

For model development, the validation subset is drawn only from the training partition. The held-out test partition is not used for gradient-based training, early stopping, hyperparameter selection, or any other model-selection step, and is accessed only for final evaluation. All compared methods are trained and evaluated using the same preprocessed inputs and the same held-out split for each dataset.

B.2 Statistical Significance Analysis

To verify that the improvements are statistically reliable, we conduct paired t-tests over 10 independent seeds between PSDNet and the second-best method on each dataset. As shown in Table 9, PSDNet achieves statistically significant gains ($p < 0.05$) on 7 out of 8 datasets, with $p < 0.01$ on 6 of them. The sole exception is mHealth, where PSDNet's marginal advantage (+0.08%) is not significant ($p = 0.55$), likely because all competing methods already saturate near 99% accuracy on this relatively simple dataset.

Table 9: Statistical significance of PSDNet's accuracy improvement over the strongest baseline on each dataset (paired t-test).(* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; n.s. = not significant.)

Dataset	Runner-up	ΔAcc	p-value	Significance
UCI	Residual	+ 0.73%	<0.001	***
USC	DeepConvContext	+ 1.77%	<0.001	***
PAMAP2	DeepConvContext	+ 1.56%	0.002	**
DSADS	DeepConvContext	+ 1.72%	0.008	**
HAPT	DeepConvContext	+ 0.70%	<0.001	***
UniMiB	CALANet	+ 2.12%	<0.001	***
MotionSense	DeepConvContext	+ 0.64%	0.043	*
mHealth	CALANet	+ 0.08%	0.545	n.s.

B.3 Complementary Analysis of Trigger Selectivity and Uncertainty Reliability

System 2 Trigger Selectivity. We further examine whether the uncertainty gate selectively fires on difficult windows under the diagnostic setting in Table 10. On HAPT and PAMAP2, the gate triggers on only 2.20% and 1.00% of windows, yet the triggered subsets exhibit System 1 error rates of 43.70% and 71.10%, more than 20 times those of the non-triggered subsets. Furthermore, 87.20% and 79.30% of triggered windows overlap activity transition boundaries, compared to 48.60% and 6.90% in non-triggered ones. On USC, the error-rate contrast is the most extreme. These results confirm that the gate preferentially targets error-prone, transition-adjacent windows rather than firing randomly.

Table 10: Triggered vs. Non-triggered Window Statistics.Trigger Rate: percentage of test windows for which the gate decides to invoke System 2. Err (triggered/non-triggered): System 1's error rate within the triggered and non-triggered subsets. If the gate is effective, triggered windows should have a much higher error rate, meaning the gate correctly identifies the samples that System 1 struggles with. Boundary (triggered/non-triggered): percentage of windows in each subset that lie at an activity transition point. A higher boundary ratio among triggered windows means the gate preferentially activates near activity changes, where recognition is most challenging.

Dataset	Trigger Rate(%)	Error Rate (triggered)(%)	Error Rate (non-triggered)(%)	Boundary(triggered)(%)	Boundary (non-triggered)(%)
HAPT	2.20 ± 0.30	43.70 ± 5.20	2.00 ± 0.30	87.20 ± 3.50	48.60 ± 0.20
PAMAP2	1.00 ± 0.80	71.10 ± 10.30	2.80 ± 0.40	79.30 ± 12.70	6.90 ± 0.60
USC	9.90 ± 2.00	47.80 ± 2.40	0.90 ± 0.70	4.30 ± 0.50	2.90 ± 0.00

Uncertainty as an Error Predictor. We assess whether the uncertainty score can independently predict System 1 failures and boundary locations(Table 11). The AUROC for predicting System 1 errors exceeds 0.92 across all three datasets, reaching 0.97 on USC, confirming that the learned uncertainty reliably identifies hard-to-classify windows. For boundary-window prediction, AUROC reaches 0.62, 0.85, and 0.69 on HAPT, PAMAP2, and USC, respectively. These results reveal dataset-dependent associations between System-1 uncertainty and activity transitions, with the strongest correspondence observed on PAMAP2.

Table 11: Uncertainty Score as a Predictor: AUROC for System-1 Error and Boundary Window Detection. AUROC (Sys1 Error): the area under the ROC curve when using the uncertainty score to predict whether System 1 misclassifies a given window. A value close to 1.0 means the uncertainty score reliably identifies windows that System 1 will get wrong. AUROC (Boundary Window): the AUROC when using the same uncertainty score to predict whether a window overlaps an activity transition boundary. A higher AUROC indicates a stronger association between System-1 uncertainty and boundary-window status.

Dataset	AUROC (Sys1 Error)	AUROC (Boundary Window)
HAPT	0.94 ± 0.01	0.62 ± 0.03
PAMAP2	0.92 ± 0.02	0.85 ± 0.01
USC	0.97 ± 0.01	0.69 ± 0.02

B.4 Complementary Accuracy-Efficiency Pareto Analysis

Accuracy-Efficiency Pareto Analysis. The Pareto analysis on the HAPT and USC datasets further corroborates the efficiency of selective deliberation (Fig. 6). On HAPT, PSDNet maintains a Weighted F1 of 97.8% across all trigger rates down to 5%, representing a 1.9-point improvement over the CNN baseline (95.9%). This indicates that the uncertainty gate is activated primarily for samples that truly benefit from System 2. On USC, PSDNet improves F1 from 94.4% to a peak of 96.2%. The curve remains stable above a 20% trigger rate and degrades only slightly to 95.6% at the 5% extreme, reflecting the higher inter-class ambiguity in this dataset. Together with the PAMAP2 results, these findings demonstrate that PSDNet consistently achieves a favorable accuracy–efficiency frontier, retaining near-optimal accuracy at a fraction of the full deliberation cost across datasets of varying difficulty.

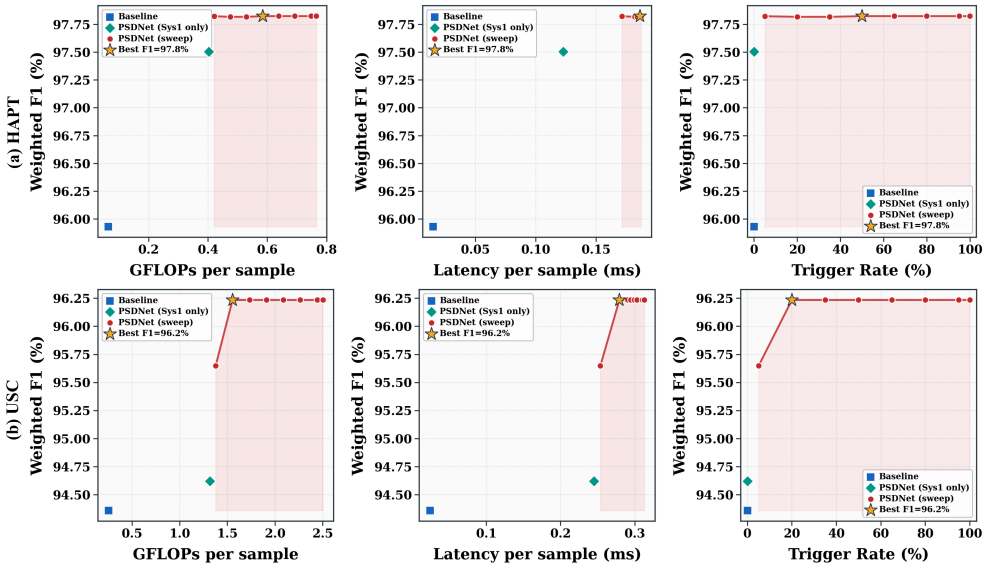


Figure 6: Accuracy–efficiency Pareto analysis of PSDNet on (a)HAPT and (b)USC under varying System 2 trigger rates. Each row shows, from left to right: Weighted F1 vs. GFLOPs per sample, Weighted F1 vs. inference latency per sample, and Weighted F1 vs. System 2 trigger rate. Blue squares: CNN baseline; teal diamonds: PSDNet (System 1 only); red circles: PSDNet with varying uncertainty thresholds; stars: best operating points. Shaded regions indicate the Pareto-dominant area.

Fairness Comparison under Matched History Access and Parameter Budgets. As shown in Table 12, PSDNet-Small achieves the best accuracy and F1 score, reaching 81.63% and 81.03%, respectively, while requiring fewer parameters and lower GFLOPs than the other context-aware baselines. It outperforms the strongest non-PSDNet baseline, CNN+HistoryMHA, by 2.35 percentage points in accuracy and 1.83 percentage points in F1. The improvement is most visible on near-boundary regions, where PSDNet-Small reaches 61.07% Near-Bnd F1, compared with 54.88% for CNN+HistoryMHA. This suggests that explicitly modeling phase-structured temporal context is useful for resolving ambiguous transitions. Moreover, the widened ContextNet-PM* baseline, despite having a parameter budget close to PSDNet, does not match the performance of PSDNet and even underperforms the lighter CNN+HistoryGRU. The UniMiB controls show that PSDNet can outperform baselines with matched history and approximately matched capacity. Table 13 further shows that PSDNet outperforms the Widened Context control on HAPT, both overall and across all reported sequence regions.

B.5 Complementary Analysis of Script-Compatibility Patterns

Fig. 7 visualizes the script-compatibility scores across all activity classes on (a) HAPT and (b) PAMAP2. In both datasets, the highest scores (darkest regions) consistently align with the ground-truth class rows, indicating that the learned scripts effectively capture class-specific temporal patterns.

Table 12: Fairness comparison on UniMiB under matched history access and parameter budgets.All methods share the same CNN backbone, training protocol, and optimizer settings.History denotes the number of past sensor windows available at inference time:0 uses only the current window w_t, whereas L=10 uses the ten preceding windows.All context-aware methods access the same history and differ only in their aggregation mechanism(e.g., GRU, multi-head attention, or phase-script matching).*The parameter budget is widened to approximate PSDNet's capacity.Results are mean ± std over 5 seeds; **bold** indicates the best result per column.

Model	History	Params	GFLOPs	Acc (%)	F1 (%)	Lat. (ms)	Boundary F1	Near-Bnd F1	Interior F1	Trigger
CNN	0	1.81M	0.165	77.18±0.31	76.31±0.35	0.87	52.11±1.71	52.78±0.99	78.55±0.32	–
CNN+HistoryGRU	L	2.34M	1.821	79.08±0.56	78.83±0.71	2.43	39.04±3.66	54.68±1.58	81.51±0.62	–
CNN+HistoryMHA	L	2.60M	1.821	79.28±0.53	79.20±0.51	3.49	48.91±1.17	54.88±1.37	81.63±0.46	–
ContextNet-PM*	L	3.40M	1.830	78.57±0.22	78.51±0.34	3.51	50.80±5.28	56.96±2.54	80.66±0.37	–
PSDNet-Small	L	2.27M	1.777	**81.63±0.18**	**81.03±0.22**	3.66	51.20±2.57	**61.07±1.81**	**83.14±0.31**	0.448
PSDNet	L	3.68M	1.833	80.33±0.67	79.85±0.52	3.37	**54.57±2.69**	59.24±1.63	81.89±0.62	0.288

Table 13: Comparison with context-matched controls on HAPT and PAMAP2. Values represent weighted F1 differences (%) between PSDNet and the corresponding control models, with positive values favoring PSDNet. HistoryMHA matches PSDNet's causal history length, while Widened Context additionally approximates its parameter count

Dataset	Control	History length	Params (PSDNet / control)	Overall	Boundary	Near-boundary	Interior
HAPT	HistoryMHA	10	2.51M / 1.04M	+0.64	−0.29	+0.20	+1.22
	Widened Context	10	2.51M / 2.23M	+1.41	+2.24	+1.16	+1.19
PAMAP2	HistoryMHA	8	4.74M / 3.27M	+0.23	+6.58	+4.04	−0.02
	Widened Context	8	4.74M / 4.46M	−0.23	−5.83	−8.65	+0.38

During stable activity segments, the score distribution is sharply peaked at the correct class, while near transition boundaries the scores briefly spread across multiple classes before converging to the new activity, reflecting the inherent ambiguity at these junctures. This confirms that the script-compatibility mechanism provides a discriminative and interpretable basis for activity classification.

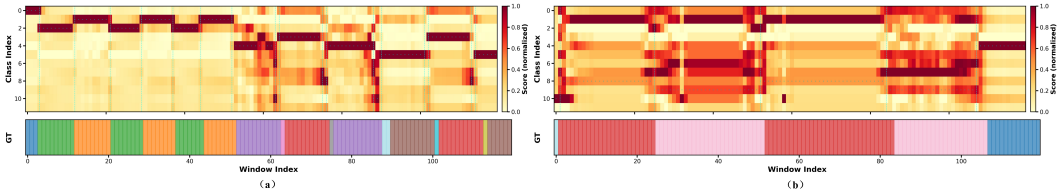


Figure 7: Script-compatibility score heatmap on (a) HAPT and (b) PAMAP2. The horizontal axis represents the window index. The upper panel displays the normalized script-compatibility scores (y-axis: class index; color intensity: score magnitude), where dashed markers indicate the ground-truth class for each window. The lower panel shows the corresponding ground-truth activity labels as color-coded bands.

Table 14: Primitive and Script Compatibility Changes around Activity Transitions. Primitive JS is the Jensen–Shannon divergence between the phase mixtures of adjacent windows, and the two ratio columns compare boundary and near-boundary divergence with interior divergence. Compatibility shift measures the change in script compatibility toward the incoming activity, so positive values favor the incoming activity.

Dataset	Boundary primitive JS	Near-boundary primitive JS	Interior primitive JS	Boundary / Interior	Near-boundary / Interior	Compatibility shift
HAPT	0.3812 ± 0.0104	0.0364 ± 0.0281	0.0196 ± 0.0099	$19.45\times$	$1.86\times$	$+6.96 \pm 2.78$
PAMAP2	0.2709 ± 0.0509	0.3022 ± 0.0159	0.0492 ± 0.0086	$5.51\times$	$6.14\times$	$+1.14 \pm 1.01$

As shown in Table 14, primitive-distribution divergence is substantially higher near activity transitions than in interior regions, with Boundary-to-Interior ratios of $19.45\times$ on HAPT and $5.51\times$ on PAMAP2. Compatibility also shifts toward the incoming activity on both datasets, clearly on HAPT and more weakly on PAMAP2.

Table 15: Chronological-History Perturbation. Shuffled randomly permutes the order of the history windows while keeping their content. F1 changes are reported in points, and negative values mean that shuffling lowers F1.

Dataset	Script Top-1: Original → Shuffled (%)	Script Margin: Original → Shuffled	System 1 F1 Change (points)	Final F1 Change (points)
HAPT	$84.82 \pm 12.74 \rightarrow 75.00 \pm 12.38$	$3.87 \pm 1.82 \rightarrow 2.53 \pm 1.40$	-1.73 ± 0.58	-1.60
PAMAP2	$57.75 \pm 20.53 \rightarrow 57.73 \pm 19.85$	$1.31 \pm 1.23 \rightarrow 1.29 \pm 1.20$	-0.12 ± 0.09	-0.21
MotionSense	$80.75 \pm 9.61 \rightarrow 80.55 \pm 9.88$	$3.52 \pm 0.89 \rightarrow 3.51 \pm 0.88$	-0.19 ± 0.11	-0.23

HAPT shows the clearest response to chronological randomization across script ranking, script margin, and prediction F1 (Table 15). The smaller changes on PAMAP2 and MotionSense indicate that dependence on ordered history varies across datasets.

B.6 Hyperparameter Sensitivity Analysis

As shown in Table 16, PSDNet exhibits stable performance across different hyperparameter settings. The history length L has a relatively more pronounced impact: when L is small, the model has insufficient temporal context, leading to degraded performance, as L increases to 8 (PAMAP2) or 10 (HAPT), the performance gains gradually saturate. The number of phase primitives M also affects the results, but the performance varies only slightly around 12 (PAMAP2) or 8 (HAPT). In contrast, the script length R has a weaker influence, indicating that PSDNet is not sensitive to the choice of script length. The candidate size K_c mainly governs the accuracy–efficiency trade-off: a smaller K_c restricts the candidate set and increases System-2 triggering, whereas $K_c = 3$ provides a favorable balance.

Table 16: Hyperparameter Sensitivity Analysis on PAMAP2 and HAPT. We evaluate the impact of history length L, the number of phase primitives M, script length R, and candidate size K_c on model performance. Results are averaged over 5 independent runs with different random seeds.

HP	Val	PAMAP2			HAPT		
		Acc(%)	**F1(%)**	**Trigger Rate(%)**	**Acc(%)**	**F1(%)**	**Trigger Rate(%)**
L	2	95.45 ± 0.49	95.42 ± 0.51	1.07 ± 0.08	96.02 ± 0.23	95.99 ± 0.23	2.08 ± 0.31
	4	96.62 ± 0.50	96.62 ± 0.51	1.26 ± 1.11	96.71 ± 0.17	96.68 ± 0.18	2.66 ± 0.77
	6	96.62 ± 0.30	96.60 ± 0.32	0.83 ± 0.12	97.10 ± 0.25	97.08 ± 0.25	1.69 ± 0.34
	8	$\mathbf{96.74 \pm 0.58}$	$\mathbf{96.73 \pm 0.59}$	0.53 ± 0.16	97.23 ± 0.20	97.21 ± 0.20	2.26 ± 0.64
	10	96.62 ± 0.39	96.61 ± 0.40	0.76 ± 0.49	$\mathbf{97.53 \pm 0.26}$	$\mathbf{97.53 \pm 0.25}$	2.65 ± 1.05
M	4	96.14 ± 0.28	96.12 ± 0.27	1.90 ± 0.96	97.45 ± 0.40	97.43 ± 0.41	5.58 ± 2.41
	8	96.82 ± 0.29	96.80 ± 0.29	1.31 ± 0.52	$\mathbf{97.60 \pm 0.12}$	$\mathbf{97.58 \pm 0.12}$	2.07 ± 0.52
	12	96.65 ± 0.35	96.62 ± 0.36	0.54 ± 0.10	97.45 ± 0.21	97.43 ± 0.22	2.15 ± 0.89
	16	$\mathbf{97.08 \pm 0.63}$	$\mathbf{97.06 \pm 0.64}$	1.74 ± 1.24	97.40 ± 0.30	97.38 ± 0.30	1.71 ± 0.40
	20	97.04 ± 0.34	97.03 ± 0.33	0.91 ± 0.51	97.36 ± 0.10	97.34 ± 0.10	1.28 ± 0.26
R	2	96.49 ± 0.38	96.47 ± 0.39	1.09 ± 0.82	97.51 ± 0.17	97.49 ± 0.17	1.64 ± 0.32
	4	96.74 ± 0.18	96.73 ± 0.18	0.46 ± 0.21	97.50 ± 0.25	97.48 ± 0.25	1.94 ± 0.54
	6	$\mathbf{96.81 \pm 0.37}$	$\mathbf{96.78 \pm 0.37}$	0.68 ± 0.35	97.44 ± 0.27	97.42 ± 0.27	1.65 ± 0.18
	8	96.70 ± 0.15	96.68 ± 0.16	0.78 ± 0.19	$\mathbf{97.65 \pm 0.23}$	$\mathbf{97.62 \pm 0.24}$	1.35 ± 0.38
K_c	1	96.67 ± 0.49	96.64 ± 0.51	0.58 ± 0.25	97.42 ± 0.26	97.39 ± 0.25	3.79 ± 1.78
	2	$\mathbf{96.97 \pm 0.48}$	$\mathbf{96.96 \pm 0.49}$	0.56 ± 0.28	97.19 ± 0.16	97.16 ± 0.16	1.79 ± 0.23
	3	96.97 ± 0.66	96.95 ± 0.67	0.92 ± 0.28	$\mathbf{97.53 \pm 0.26}$	$\mathbf{97.50 \pm 0.25}$	2.15 ± 0.28
	4	96.89 ± 0.46	96.88 ± 0.46	0.61 ± 0.34	97.52 ± 0.11	97.50 ± 0.11	1.88 ± 0.41
	5	96.66 ± 0.22	96.65 ± 0.22	0.60 ± 0.29	97.50 ± 0.25	97.49 ± 0.26	1.72 ± 0.28

B.7 Segmentation Robustness

Tables 17 and 18 show that PSDNet maintains a positive weighted-F1 margin over HistoryMHA across all evaluated window configurations and grid offsets. Relative to Native, the Short and Sparse

settings reduce PSDNet's F1 by only 0.03 and 0.57 points, respectively, compared with 0.37 and 0.99 points for HistoryMHA. The gap is significant at $p < 0.05$ in four of the six distinct settings, while the Sparse configuration and the 0.25 offset show the same direction without reaching significance. These results indicate that PSDNet remains stable across different window constructions and alignments on MotionSense.

Table 17: Window-configuration sensitivity on MotionSense. Window length W and stride S are in samples. The look-back horizon $LS/50$ is 7.68 seconds for every setting. Entries are mean $\pm$ standard deviation of weighted F1 (%); Gap is PSDNet minus HistoryMHA in percentage points. The p values are from two-sided paired tests within each setting.

Dataset	Setting	W	S	Overlap	L	PSDNet	HistoryMHA	Gap (pp)	p
MotionSense	Native	128	64	50%	6	97.15 ± 0.16	96.76 ± 0.31	$+0.39$	0.009
MotionSense	Short	64	32	50%	12	97.12 ± 0.20	96.39 ± 0.22	$+0.73$	0.008
MotionSense	Sparse	128	128	0%	3	96.58 ± 0.41	95.77 ± 0.25	$+0.81$	0.069

Table 18: Test-time segmentation-grid offsets on MotionSense. Offsets are fractions of the native 64-sample stride. Entries are mean $\pm$ standard deviation of weighted F1 (%); Gap is PSDNet minus HistoryMHA in percentage points. The p values compare the two methods on the same windows within each offset.

Dataset	Offset / stride	Samples	Test windows	PSDNet	HistoryMHA	Gap (pp)	p
MotionSense	0.00	0	4,352	97.15 ± 0.16	96.76 ± 0.31	$+0.39$	0.009
MotionSense	0.25	16	4,336	97.02 ± 0.21	96.72 ± 0.26	$+0.30$	0.065
MotionSense	0.50	32	4,313	96.93 ± 0.17	96.62 ± 0.20	$+0.31$	0.024
MotionSense	0.75	48	4,295	96.98 ± 0.17	96.60 ± 0.37	$+0.38$	0.046

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: The claims in the abstract and introduction are consistent with the experimental results and theoretical analysis, accurately describing the scope and limitations of the paper's contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper explicitly discusses the assumptions, scope, and potential shortcomings of the method in a dedicated Limitations section.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All theorems are fully proven in the appendix, with intuitive sketches provided in the main text.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper describes model architecture, hyperparameter settings, and data processing pipelines in detail in the main text and appendix.

   Guidelines:

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

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

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

Answer: [No]

Justification: The experiments are conducted on publicly available HAR benchmark datasets, and the paper provides references and preprocessing details for accessing and preparing the data. The implementation code is not publicly released at submission time to preserve anonymity, but we plan to make it publicly available after the paper is published, together with environment setup and execution instructions.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Experimental settings including data splits, learning rate, and optimizer are detailed in the appendix.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: All main results report mean and standard deviation across multiple random seed runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The appendix includes the details on the computer resources.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and the research complies in all respects.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This work can positively impact downstream applications such as wearable sensing, mobile health monitoring, smart home systems, elderly care, and human-centered activity analysis.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The models/datasets released in this work do not pose high-risk misuse scenarios.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The creators and original owners of all datasets used in this work are properly cited in the references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets such as datasets, model checkpoints, or code at submission time. The implementation code is planned to be made publicly available after the paper is published, together with accompanying documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This research does not involve human subjects and therefore does not require IRB review.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method of PSDNet is designed for sliding-window human activity recognition using sensor-based time-series data. It relies on phase primitives, recent temporal history, uncertainty-aware deliberation, and class-specific phase scripts, with no LLMs involved as a methodological component.

    Guidelines:

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