



UASD-Edge: Uncertainty-Aware Self-Distillation for Early Human Activity Recognition on Edge Devices

Antti Saarinen¹, Laura Niemi¹, Matthias Peeters², Jonas Verhoeven^{2*}

¹ Computer Science and Engineering Research Unit, University of Oulu, Oulu, Finland

² Department of Computer Science, University of Antwerp, Antwerp, Belgium

*Co-corresponding authors: Jonas Verhoeven, jonas2verhoeven@uantwerpen.be

Abstract

Early human activity recognition (HAR) on edge devices must trade classification quality against sensing delay while avoiding unreliable supervision from incomplete observations. This paper presents UASD-Edge, a compact early-HAR framework that augments full-window-to-prefix self-distillation with per-sample teacher reliability. The reliability is computed from normalized entropy of the detached full-window prediction and weights only the distillation term, leaving hard-label supervision unchanged. A single 7,995-parameter temporal CNN is shared by all prefix lengths, and no uncertainty branch or additional inference parameters are introduced. We evaluate on the official subject-disjoint UCI Human Activity Recognition Using Smartphones split using the three dynamic locomotion classes and six inertial sequence channels. Across seeds 13, 37, and 73, UASD-Edge obtains macro-F1 of $94.51 \pm 1.18\%$, $97.88 \pm 1.12\%$, $98.22 \pm 1.07\%$, and $98.56 \pm 0.77\%$ at 25%, 50%, 75%, and 100% observation. Against a same-backbone PrefixKD baseline with the identical distillation coefficient, UASD is lower by 0.22 percentage points at 25% but improves the later three checkpoints by 0.14–0.19 points, so no statistical-significance claim is made. A validation-selected global MSP gate reaches $98.05 \pm 0.75\%$ test accuracy and $98.03 \pm 0.76\%$ macro-F1 after observing $32.59 \pm 2.96\%$ of the 2.56-s window on average (0.834 s). Frozen TorchScript batch-1 FP32 inference takes 0.0319 ms median at the full window on one Intel Xeon Platinum 8573C CPU thread. The results support uncertainty-weighted distillation as a low-cost regularizer for later prefixes while exposing a reproducible accuracy-earliness operating point.

Keywords: early human activity recognition, self-distillation, predictive uncertainty, edge intelligence, wearable sensing, early time-series classification.

1. Introduction

Wearable and mobile sensing systems increasingly need a useful decision before a complete sensing window is available. In activity-aware interfaces, gait assistance, safety monitoring, and adaptive sensor scheduling, waiting for an entire segment may dominate the response time even when the neural network itself is fast. Wearable HAR has mature classical and deep-learning foundations [2]–[9], yet the standard full-window benchmark does not reveal how performance evolves as evidence arrives. Early time-series classification instead treats earliness and correctness as coupled objectives [10]–[12]. A particularly strong deployment-oriented point of departure is PrefixKD-Edge [1], which makes prefix-level supervision explicit while also measuring physical observation delay and runtime cost. We retain this practical premise—one compact backbone should serve multiple observation lengths—but focus on a different question: when the full-window prediction is itself uncertain for a particular sample, should it supervise an early prefix as strongly as a confident full-window prediction? Plain self-distillation applies the same global coefficient to every soft target, although teacher reliability can vary substantially from sample to sample.

This question matters on edge devices for two reasons. First, uncertainty estimators based on Monte Carlo dropout or deep ensembles are effective but require repeated stochastic passes or multiple models [17], [18], which can conflict with a tight inference budget. Second, early-exit and selective-classification methods require a confidence signal whose operational meaning should be evaluated rather than assumed [19], [20]. We therefore use a deterministic uncertainty signal already available from the full-window softmax during training: normalized predictive entropy. It costs no extra inference computation because it is used only to scale the training-time knowledge-distillation term. The resulting method, UASD-Edge, makes four contributions. First, it introduces per-sample entropy-based reliability for full-to-prefix self-distillation while preserving hard-label cross-entropy for every sample. Second, it uses exactly the same 7,995-parameter CNN, distillation temperature, and global distillation coefficient as the plain PrefixKD comparison, enabling clean attribution to uncertainty weighting. Third, it couples prefix training with validation-only temperature calibration and a transparent global confidence gate, reporting both static checkpoint quality and adaptive decision time. Fourth, it reports three-seed test results, stored predictions, calibration metrics, model size, measured one-thread CPU latency, and a clearly labeled standardized-input corruption diagnostic; negative results at the shortest prefix are retained rather than hidden.

II. RELATED WORK

A. Wearable HAR and Compact Temporal Models

Wearable HAR has been surveyed extensively with emphasis on sensors, placement, segmentation, features, and evaluation protocols [2], [3]. Deep convolutional models learn discriminative patterns directly from multichannel sensor sequences [4], while convolutional-recurrent systems [5], smartphone deep networks [6], recurrent HAR models [8], real-time accelerometer CNNs [9], and the DeepSense framework [7] established strong end-to-end alternatives to handcrafted pipelines. Temporal convolutional networks also provide a parallel sequence-modeling family with long effective receptive fields [21]. These works motivate lightweight temporal learning, but most report a prediction after a complete input window.

B. Early Classification and Selective Decisions

Early classification seeks a label from a partial time series while explicitly accounting for the cost of waiting. Xing et al. formulated early time-series classification using discriminative prefixes [10]; Mori et al. jointly optimized accuracy and earliness [11]; and TEASER combined accurate time-series classification with a dedicated early decision mechanism [12]. In deep networks, BranchyNet attaches intermediate exits to reduce average computation [20], whereas selective classification allows abstention when confidence is insufficient [19]. Our setting differs from architectural early exits: the same temporal backbone processes each available prefix, so inference parameters do not grow with the number of decision checkpoints.

C. Distillation, Uncertainty, and Calibration

Knowledge distillation transfers class-distribution information from a teacher to a student [13], and self-distillation uses predictions or representations from the same model family to regularize learning [14]. Self-knowledge distillation has also been investigated for wearable HAR, including TASKED for representation learning under sensor-domain variability [15]. Building on the clear systems formulation of [1]—and its explicitly identified opportunity to extend prefix training with uncertainty-aware distillation—we ask whether teacher confidence can control how strongly future evidence supervises a short prefix. Unlike Bayesian dropout [17] or ensembles [18], our reliability estimate is a single deterministic entropy computed from a training-time full-window prediction. Because modern networks can be miscalibrated, we separately apply validation temperature scaling [16] before using confidence for the deployment stopping rule; training-time uncertainty weighting and test-time stopping are therefore not conflated. Edge deployment additionally requires attention to actual runtime behavior. MCUNet [22], TensorFlow Lite Micro [23], MicroNets [24], and integer-only quantization work [25] show that operator support, memory planning, quantization, and measured platform latency matter beyond nominal parameter count. We therefore report a server-CPU implementation proxy and avoid interpreting it as a microcontroller energy or peak-RAM measurement.

III. PROBLEM FORMULATION

Let $X=[x_1,...,x_T]$ denote a six-channel inertial window and y in $\{1,2,3\}$ its activity label. The original UCI sequence contains $T=128$ samples at 50 Hz. A non-overlapping causal four-sample block mean produces a 32-step sequence while preserving the 2.56-s physical duration. Let $R=\{0.25,0.50,0.75,1.00\}$; at checkpoint

r, only the first Lr in {8,16,24,32} reduced-rate steps are visible, corresponding to 0.64, 1.28, 1.92, and 2.56 s. No sample after the selected prefix is supplied to the network.

For static evaluation, accuracy, macro-F1, negative log-likelihood (NLL), 15-bin expected calibration error (ECE), Brier score, and normalized entropy are reported independently at each checkpoint. For adaptive inference, a confidence score sr is evaluated in chronological order and the first checkpoint satisfying $sr \geq \tau$ is returned; otherwise the complete window is used. We test maximum softmax probability (MSP) and one minus normalized predictive entropy. The stopping policy changes sensing/decision delay, not the parameter count of the classifier.

IV. UASD-EDGE METHOD

A. Causal Front End and Shared TinyCNN

The input consists of body acceleration on three axes and gyroscope signals on three axes. Four consecutive samples are averaged causally, reducing the temporal length from 128 to 32. Channel-wise mean and standard deviation are estimated only on the seed-specific training subset. The backbone applies 1-D convolutions 6->24 (kernel 5), 24->32 (kernel 5), a length-2 max pool, 32->32 (kernel 3), batch normalization and ReLU after each convolution, global temporal average pooling, and a three-class linear head. The PyTorch implementation contains 7,995 trainable parameters. Symmetric convolution padding is applied only inside the supplied prefix, so no unobserved future sample is introduced.

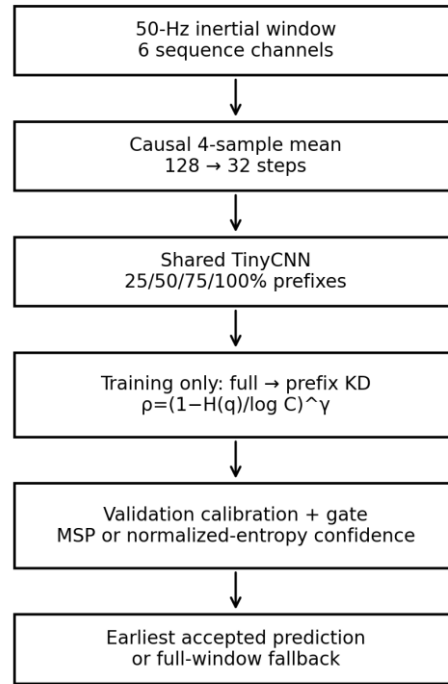


Figure 1 UASD-Edge training and adaptive-inference pipeline. The entropy reliability is training-only; deployment uses the shared TinyCNN and a calibrated confidence gate

B. Stochastic Multi-Prefix Supervision

Evaluating every prefix during every training update would multiply the number of forward/backward paths. Instead, each mini-batch always uses the full sequence and uniformly samples one early ratio from {0.25,0.50,0.75}. Thus all early checkpoints receive direct supervision across epochs while each batch requires two forward passes. With z_f and z_p denoting full and sampled-prefix logits, the base objective is $L_{CE} = 0.5 CE(z_f, y) + 0.5 CE(z_p, y)$. (1)

C. Uncertainty-Aware Full-to-Prefix Self-Distillation

The full window contains evidence that is unavailable to the prefix, but its prediction is not equally reliable for every sample. Let $q = \text{softmax}(z_f)$ with z_f detached from the prefix gradient. We define normalized entropy $h(q) = -\sum_c q_c \log(q_c) / \log(C)$, where $C=3$, and convert it to a reliability weight

$$\rho = [1 - h(q)]^\gamma, \quad \gamma = 2. \quad (2)$$

For distillation temperature $T_d=2$, $q_T=\text{softmax}(z_f/T_d)$ and $p_T=\text{softmax}(z_p/T_d)$. UASD weights only the per-sample KL term; the hard-label loss in (1) is never reduced for an uncertain sample:

$$L_{\text{UASD}} = L_{\text{CE}} + \alpha T_d^2 (1/B) \sum_i \rho_i \text{KL}(q_i^{T_d} || p_i^{T_d}). \quad (3)$$

where $\alpha=0.35$ and B is the mini-batch size. The PrefixKD comparison uses the identical equation with $\rho_i=1$ for all samples. This same-alpha design isolates the effect of teacher reliability instead of confounding it with stronger or weaker global regularization. Detachment prevents the prefix term from moving its teacher target within the same update. At inference, the full-window branch is not a second model; it is simply the same network evaluated when more observations are available.

D. Calibration and Transparent Stopping

After model selection, one scalar temperature is fitted per seed by minimizing validation NLL over logits pooled from all four prefixes. Accuracy and macro-F1 are unchanged because temperature scaling preserves the logit argmax. For each method and confidence score, a single threshold τ shared across the three seeds is then selected by sweeping 0.400 to 0.995 in increments of 0.001 on validation predictions. Among thresholds whose mean adaptive validation accuracy is no more than 0.05 percentage points below mean full-window validation accuracy, we choose the threshold with minimum mean observation ratio, breaking ties by accuracy and macro-F1. The threshold is then frozen and applied to the held-out test subjects.

V. EXPERIMENTAL PROTOCOL

A. Dataset and Split

Experiments use the public Human Activity Recognition Using Smartphones dataset [26], [27]. It contains waist-mounted smartphone inertial recordings from 30 volunteers, sampled at 50 Hz and segmented into 2.56-s windows. We preserve the official subject-disjoint development/test partition and retain only WALKING, WALKING_UPSTAIRS, and WALKING_DOWNSTAIRS. This yields 3,285 development windows and 1,387 held-out test windows. The six supplied sequence channels are used rather than the 561 engineered features. For seeds 13, 37, and 73, the development set is split stratifiably into 2,792 training and 493 validation windows; test subjects are never used for normalization, checkpoint selection, temperature fitting, or threshold selection.

Table 1 DATASET AND EVALUATION PROTOCOL

Item	Setting
Dataset	UCI HAR smartphones; 3 dynamic classes
Dev / test	3,285 / 1,387 windows
Input	6 channels; 128 samples at 50 Hz
Front end	causal 4-sample mean $\rightarrow$ 32 steps
Checkpoints	0.64 / 1.28 / 1.92 / 2.56 s
Seeds	13, 37, 73
Backbone	7,995-parameter shared TinyCNN

B. Baselines and Attribution

Multi-prefix CE uses the same stochastic two-pass training scheme but no distillation. PrefixKD uses full-to-prefix KL with $\alpha=0.35$ and $T_d=2$, with no uncertainty weighting. UASD uses the same α , temperature, architecture, input preprocessing, optimizer, splits, and selection criterion; the sole main-comparison loss change is ρ in (3). All reported baseline numbers are generated by our experimental runs rather than transcribed from previous work.

C. Optimization and Model Selection

Models are trained in PyTorch using AdamW [28] with learning rate $1e-3$, weight decay $1e-4$, batch size 256, gradient-norm clipping at 5, and cosine annealing for at most 30 epochs. The cosine schedule follows the SGDR family [29]. Early stopping patience is six epochs, and the selected checkpoint maximizes validation macro-F1 averaged over all four observation ratios. The implementation uses PyTorch [30]. We report the

arithmetic mean and sample standard deviation across the three optimization seeds and do not use three runs to claim statistical significance.

D. Deployment and Diagnostic Measurements

For the seed-13 UASD checkpoint, batch-1 FP32 latency is measured with one Intel Xeon Platinum 8573C CPU thread in both PyTorch eager mode and frozen TorchScript. Each prefix receives 200 warm-up calls and 2,000 timed calls; median and 95th-percentile latency are reported. The software environment is Python 3.13.5 and PyTorch 2.10.0+cpu. The serialized FP32 state dictionary occupies 39.32 KiB. These are reproducible software-stack indicators, not direct microcontroller energy, integer-quantized latency, or peak activation-memory results. As a limited diagnostic, the seed-13 standardized test input is additionally perturbed with zero-mean Gaussian noise of standard deviation 0.15 and 0.30 using a fixed random seed. This probe is intentionally not presented as a physical sensor-noise model; it only tests whether predictive entropy reacts in the expected direction under controlled input degradation.

VI. RESULTS

A. Static Prefix Recognition

Table 2 TEST MACRO-F1 (%) ACROSS OBSERVATION RATIOS (MEAN $\pm$ SD, THREE SEEDS)

Method	25%	50%	75%	100%
Multi-prefix CE	93.94 ± 1.59	97.70 ± 1.33	97.97 ± 1.21	98.32 ± 1.02
PrefixKD	94.73 ± 1.27	97.73 ± 1.10	98.03 ± 1.07	98.44 ± 0.76
UASD-Edge	94.51 ± 1.18	97.88 ± 1.12	98.22 ± 1.07	98.56 ± 0.77

UASD-Edge reaches $94.51 \pm 1.18\%$ macro-F1 after only 0.64 s and $98.56 \pm 0.77\%$ at the full 2.56-s window. Relative to the same-backbone PrefixKD baseline, the uncertainty weighting is not uniformly beneficial: mean macro-F1 decreases by 0.22 percentage points at 25% observation, then increases by 0.14, 0.19, and 0.12 points at 50%, 75%, and 100%, respectively. Averaged across all four checkpoints, UASD improves macro-F1 from 97.23% to 97.29%. The small magnitude and three-seed design do not justify a significance claim. The shortest-prefix loss is an important negative result: a highly uncertain full-window teacher can be downweighted so strongly that the soft-target regularizer contributes less exactly where early discrimination is hardest.

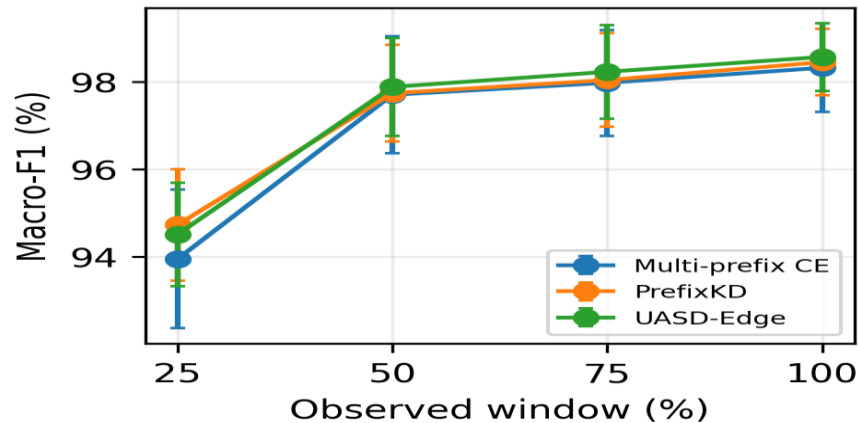


Figure 2 Test macro-F1 versus observed window. Error bars are sample standard deviations over seeds 13, 37, and 73.

B. Calibration Behavior

Temperature scaling yields mixed but informative calibration behavior. Compared with PrefixKD, UASD has lower mean NLL at 50% (0.0561 versus 0.0570), 75% (0.0557 versus 0.0574), and 100% (0.0458 versus 0.0477), but higher NLL at 25% (0.1874 versus 0.1821). ECE is likewise lower for UASD at 50% (0.0095 versus 0.0111) and 75% (0.0100 versus 0.0108), while PrefixKD is better at 25% and slightly better at the full

window. Thus the method should not be described as a universal calibration improvement; its strongest effect is a modest later-prefix regularization gain.

C. Adaptive Decision Timing

Table 3 VALIDATION-SELECTED GLOBAL GATES ON THE HELD-OUT TEST SET

Method/score	tau	Acc. (%)	F1 (%)	Obs. (%)
CE/entropy	0.929	97.91±0.97	97.88±0.99	32.67±2.55
CE/msp	0.987	97.91±1.04	97.88±1.06	32.81±2.57
PrefixKD/entropy	0.947	98.01±0.69	97.99±0.70	32.91±3.07
PrefixKD/msp	0.991	98.01±0.69	97.99±0.70	33.07±3.10
UASD/entropy	0.939	98.03±0.73	98.01±0.73	32.46±2.98
UASD/msp	0.989	98.05±0.75	98.03±0.76	32.59±2.96

The most favorable UASD operating point uses MSP with $\tau=0.989$: $98.05\pm0.75\%$ accuracy and $98.03\pm0.76\%$ macro-F1 at $32.59\pm2.96\%$ mean observation. This corresponds to 0.834 s of the 2.56-s window and a 67.41% reduction in mean sensing delay relative to always waiting for the complete window. Entropy confidence gives a similar result— $98.03\pm0.73\%$ accuracy at $32.46\pm2.98\%$ observation. Importantly, these thresholds are selected only on validation predictions and then applied unchanged to the held-out test subjects. In these runs, adaptive accuracy remains below the corresponding static full-window accuracy because some samples are deliberately emitted early. The result should therefore be read as a measured accuracy-earliness operating point, not as evidence that later sensing is harmful.

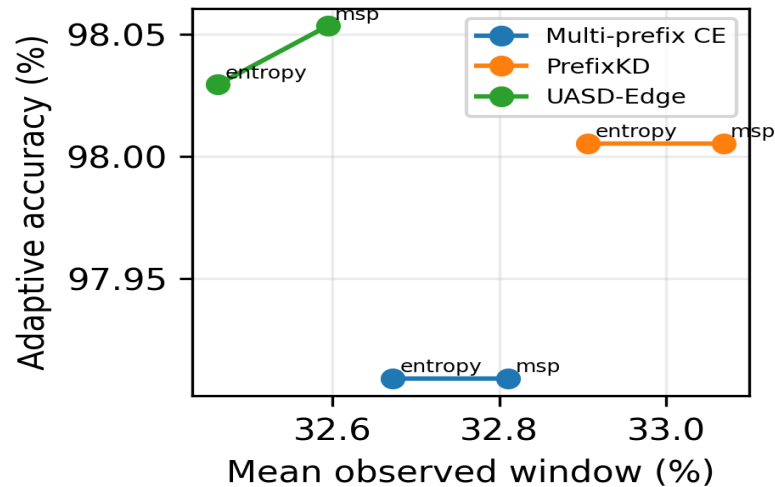


Figure 3. Adaptive test accuracy versus mean observed window for validation-selected entropy and MSP gates

D. Model Size and Measured Runtime

Table 4 FULL-WINDOW DEPLOYMENT PROXY (BATCH 1, FP32, ONE CPU THREAD)

Item	Measured value
Parameters	7,995
FP32 state dict	39.32 KiB
Eager median / P95	0.0663 / 0.0983 ms
TorchScript median / P95	0.0319 / 0.0372 ms

Frozen TorchScript median latency is 0.0310, 0.0316, 0.0325, and 0.0319 ms at 25%, 50%, 75%, and 100%,

respectively; the small non-monotonicity is within ordinary host-runtime variation at this sub-millisecond scale. UASD, PrefixKD, and multi-prefix CE have identical inference architectures, so uncertainty weighting adds no deployment parameters or operators. The absolute numbers should be interpreted only for the measured CPU/software stack.

E. Error Structure and Controlled Corruption Probe

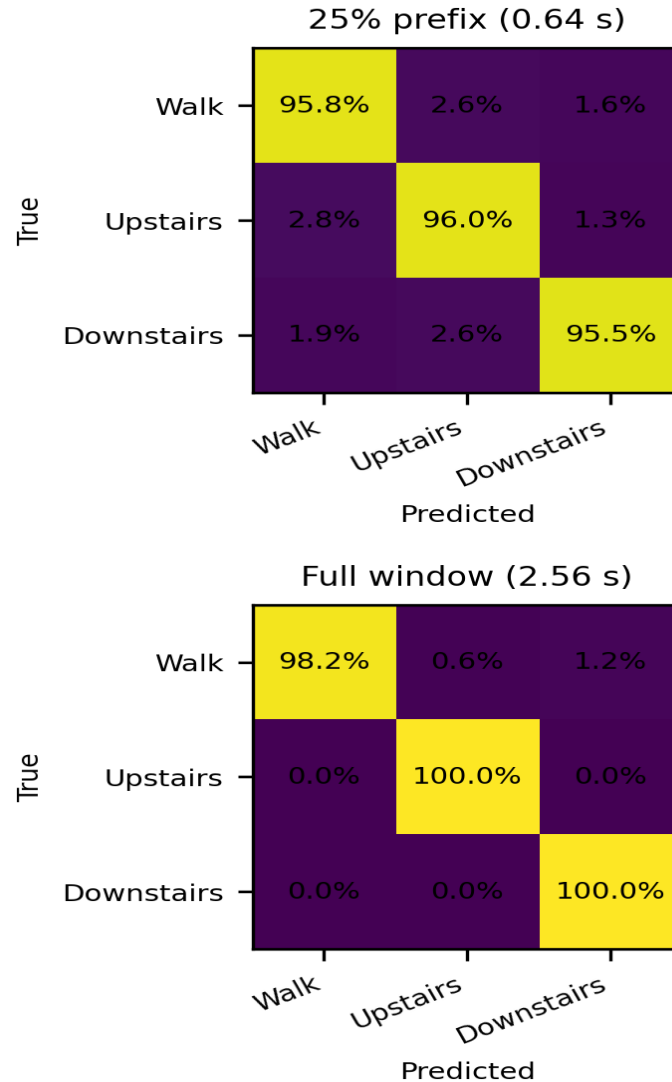


Figure 4 Row-normalized UASD-Edge confusion matrices for seed 13. The representative seed reaches 95.8%, 96.0%, and 95.5% class recall at the 25% prefix for walking, upstairs, and downstairs, respectively.

For seed 13, 25% errors are distributed across all three locomotion classes rather than concentrated in a single failed class. At the full window, upstairs and downstairs are perfect for this seed, while walking recall is 98.2%; this representative run should not be mistaken for the three-seed expectation. Under the standardized-input corruption diagnostic, 25% accuracy changes from 95.75% ($\sigma=0$) to 95.75% (0.15) and 94.38% (0.30), while normalized entropy increases from 0.0399 to 0.0459 and 0.0514. At the full window, accuracy changes from 99.35% to 99.06% and 98.92%, while entropy rises from 0.0154 to 0.0165 and 0.0214. The direction is consistent with uncertainty increasing under controlled degradation, but one seed and an artificial perturbation cannot establish real-world robustness.

F. Seed-Level Attribution and Coefficient Sensitivity*Table 5 SAME-ALPHA 75% PREFIX ATTRIBUTION BY SEED*

Seed	PrefixKD F1	UASD F1	Delta (pp)
13	98.99	99.06	+0.07
37	96.87	97.01	+0.14
73	98.24	98.60	+0.36
Mean	98.03	98.22	+0.19

At the 75% checkpoint, UASD improves over PrefixKD in all three optimization runs: +0.07, +0.14, and +0.36 percentage points for seeds 13, 37, and 73. This seed-wise direction is encouraging, but it does not convert three correlated experimental repetitions on one fixed test set into a significance claim. The purpose of the table is narrower: it shows that the mean +0.19-point result is not produced by a single favorable seed. We also executed a coefficient sensitivity run with UASD $\alpha=0.70$ while keeping the three seeds and the rest of the setup unchanged. Its macro-F1 is $94.64 \pm 1.63\%$, $97.86 \pm 1.13\%$, $98.25 \pm 1.01\%$, and $98.49 \pm 0.88\%$ from 25% to 100% observation. Relative to the main $\alpha=0.35$ UASD setting, the changes are approximately +0.13, -0.02, +0.03, and -0.07 percentage points. The stronger coefficient therefore provides no consistent advantage. We retain $\alpha=0.35$ for the main paper because matching PrefixKD's global coefficient gives the cleanest attribution to per-sample uncertainty weighting.

G. Exit Distribution and Gate Behavior*Table 6 UASD/MSP TEST DECISION DISTRIBUTION (%)*

Seed	25%	50%	75%	100%
13	89.11	6.92	1.37	2.60
37	80.32	12.18	1.73	5.77
73	76.57	13.70	2.16	7.57
Pooled	82.00	10.93	1.75	5.31

The UASD/MSP operating point is not a simple 25%-or-full switch. Pooling the three held-out test runs, 82.00% of predictions exit at 25% observation, 10.93% at 50%, 1.75% at 75%, and 5.31% fall back to the full window. These proportions reproduce the 32.59% mean observation ratio: the intermediate 50% checkpoint handles a non-trivial set of cases that are not sufficiently confident at 0.64 s, while only a small minority require 2.56 s. The full-window fallback rate also varies across seeds—from 2.60% to 7.57%—which is consistent with the observed optimization variability and reinforces the need to report more than a single favorable run. Comparing the two distillation methods under the same MSP selection procedure further limits the claim. PrefixKD reaches $98.01 \pm 0.69\%$ adaptive accuracy at $33.07 \pm 3.10\%$ observation, whereas UASD reaches $98.05 \pm 0.75\%$ at $32.59 \pm 2.96\%$. The joint change is favorable in mean accuracy and earliness but very small relative to run-to-run variation. Moreover, UASD entropy gating gives nearly the same operating point as MSP. Thus the main evidence for the proposed training rule comes from the controlled same-alpha static comparison and the retained per-seed outputs; the stopping policy should be viewed as a deployment layer that exposes useful trade-offs rather than as a mechanism used to amplify a tiny model difference.

VII. DISCUSSION

The experiments support a bounded interpretation of uncertainty-aware self-distillation. The proposed reliability weight does not transform early HAR at the shortest prefix; in fact, the plain PrefixKD baseline is

slightly better at 25%. Instead, UASD shows a small and consistent-in-mean advantage at 50–100% observation while keeping inference cost unchanged. This pattern is plausible because a teacher becomes more trustworthy as the full window resolves ambiguous motion evidence, whereas an aggressively downweighted teacher provides limited additional regularization when the sample remains uncertain. The method is therefore best viewed as a low-cost mechanism for suppressing questionable soft targets rather than as a guaranteed accuracy booster. The adaptive results are more practically consequential than the small static F1 difference. A single validation-derived threshold allows the same network to emit many decisions after roughly one third of the sensing window while preserving approximately 98% held-out accuracy. The confidence gate is deliberately simple and auditable. However, the selected confidence value is an empirical operating point rather than a safety guarantee. Applications with asymmetric risk should use task-specific validation, potentially class-conditional or conformal decision rules, and a conservative fallback rather than transplanting $\tau=0.989$ unchanged. From an edge-systems perspective, the experiment also separates sensing delay from compute latency. The reference implementation recomputes the network at each fixed checkpoint, so the 67.41% observation reduction is not claimed as a 67.41% cumulative compute or energy reduction. A production design could cache convolutional state, schedule only the next required checkpoint, or use an event-triggered front end. Quantization could further reduce storage and improve embedded efficiency [25], but int8 behavior must be measured on the target runtime rather than inferred from FP32 CPU results.

VIII. LIMITATIONS AND REPRODUCIBILITY

The study has five main limitations. First, UCI HAR is a relatively small benchmark with fixed waist placement, pre-segmented ongoing activities, and overlap between adjacent windows. The task is therefore early recognition of the current locomotion mode, not anticipation before movement onset. Second, only three optimization seeds are reported; they reveal variability but are inadequate for a well-powered significance test of sub-percentage gains. Third, the same validation subset is reused for checkpoint selection, temperature fitting, and gate selection within each seed, which is test-clean but can still make the operating point optimistic relative to a separately reserved calibration cohort. Fourth, the runtime host is a server-class CPU; no claim is made about MCU energy, peak RAM, or integer latency. Fifth, the Gaussian corruption diagnostic is synthetic and single-seed, so it is evidence only of controlled sensitivity, not of robustness to device placement, sensor drift, or real noise. To make these boundaries auditable, the reproducibility archive stores exact checkpoints for all nine main runs, seed-level validation/test logits, training histories, aggregate tables, a data-file SHA-256 manifest, environment metadata, latency outputs, the diagnostic CSV, and figure-generation code. The training script contains no hard-coded paper results. A downloader/preparation script targets the official UCI archive; the execution environment used for the reported run consumed equivalent public UCI text files and verified the development/test dimensions and class counts before training.

IX. CONCLUSION

UASD-Edge extends prefix self-distillation for early HAR with a deterministic, per-sample teacher-reliability weight derived from full-window predictive entropy. On the real UCI HAR smartphone sequences, the method preserves a 7,995-parameter shared CNN and introduces no inference-time uncertainty branch. Across three seeds it improves the same-backbone PrefixKD mean macro-F1 at 50%, 75%, and 100% observation, while losing 0.22 percentage points at 25%; the paper therefore reports a modest regularization effect rather than a universal advantage. More importantly, a validation-selected confidence gate achieves $98.05 \pm 0.75\%$ test accuracy after 0.834 s mean observation, and the full model executes in 0.0319 ms median TorchScript time on the measured CPU. The central systems lesson is that early edge HAR should optimize teacher reliability, prefix discrimination, calibrated stopping, and measured deployment cost together, while preserving negative results and separating sensing latency from runtime latency.

REFERENCES

Zhang, S. and Qiu, L. (2024). Data-driven assessment of pavement performance and resilience under extreme weather using the Long-Term Pavement Performance Program. *AI and Data Science Journal*, vol. 1, no. 1, pp. 57–66

- Zhang, H. (2024). Graph-Aware Multi-Task Learning For Early Prediction Of Timing Violations And Routing Congestion In Digital Ic Physical Design. *World Journal of Information Technology*, 2(1): 13-20.
- Wang, J., Tan, Y., Jiang, B., Wu, B., & Liu, W. (2025). Dynamic marketing uplift modeling: A symmetry-preserving framework integrating causal forests with deep reinforcement learning for personalized intervention strategies. *Symmetry*, 17(4), 610.
- Zhao, X., Liu, J., Wang, Y., & Wang, J. (2026). CryptoMamba-SSM: Linear complexity state space models for cryptocurrency volatility prediction. *IEEE Open Journal of the Computer Society*, 7, 226-243.
- Wang, J., Liu, J., Zheng, W., & Ge, Y. (2025). Temporal heterogeneous graph contrastive learning for fraud detection in credit card transactions. *IEEE Access*.
- Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. *IEEE Access*, 12, 116546-116557.
- Zhang, H. (2025). Reinforcement Learning Approaches for Layout Optimization in Electronic Design Automation with Electromagnetic Compatibility Constraints. *Frontiers in Robotics and Automation*, 2(2), 77-93.
- Han, X., Yang, Y., Chen, J., Wang, M., & Zhou, M. (2025). Symmetry-aware credit risk modeling: A deep learning framework exploiting financial data balance and invariance. *Symmetry*, 17(3), 341.
- Chen, J., Cui, Y., Zhang, X., Yang, J., & Zhou, M. (2024). Temporal convolutional network for carbon tax projection: A data-driven approach. *Applied Sciences*, 14(20), 9213.
- Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. *Asian Business Research Journal*, 10(11), 1-13.
- Chen, J., Wang, M., & Sun, T. (2025). Intelligent tax systems and the role of natural language processing in regulatory interpretation. *American Journal of Machine Learning*, 6(4), 74-94.
- Chen, Z., Liu, J., & Chen, J. (2025). Machine Learning Methods for Financial Forecasting in Enterprise Planning: Transitioning from Rule-Based Models to Predictive Analytics. *Frontiers in Artificial Intelligence Research*, 2(3), 541-564.
- Chen, J., Liu, J., Liang, Y., & Zhou, M. (2026). KE-MLLM: A knowledge-enhanced multi-sensor learning framework for explainable fake review detection. *Applied Sciences*, 16(6), 2909.
- Yue, X., Yang, J., & Liu, W. (2026). FraudDebate-Agent: A Multi-Agent LLM Framework with an Evidence-Based Debate Mechanism for Financial Statement Fraud Detection. *Mathematics*, 14(15), 2695.
- Zhu, R. & Yue, X. (2024). An Explainable Machine Learning Framework For Predicting Dividend Sustainability And Financial Risk Of U.s. Reits Under Interest-Rate Shocks. *Journal of Trends in Finance and Economics*, 1(2): 48-57.
- Yue, X. & Zhu, R. (2024). An Explainable Hybrid Machine Learning Framework For Cash-Flow-Conditioned Multi-Horizon Liquidity Risk Early Warning: An Sme-Oriented Benchmark Study. *World Journal of Management Science*, 2(1): 63-72.

- Cui, Y., Han, X., Chen, J., Zhang, X., Yang, J., & Zhang, X. (2025). FraudGNN-RL: a graph neural network with reinforcement learning for adaptive financial fraud detection. *IEEE Open Journal of the Computer Society*, 6, 426-437.
- Zi, B. (2024). Cloud-native distributed systems for real-time payment intelligence. *AI and Data Science Journal*, 1(1), 51-56.
- Zhang, S. & Qiu, L. (2023). An Interpretable, Uncertainty-Aware Framework For Bridge Deterioration Prediction And Risk-Based Maintenance Prioritization. *Academic Journal of Architecture and Civil Engineering*, 1(2): 21-28.
- Jin, J., Xing, S., Ji, E., & Liu, W. (2025). Xgate: explainable reinforcement learning for transparent and trustworthy api traffic management in iot sensor networks. *Sensors*, 25(7), 2183.
- Xing, S., Wang, Y., & Liu, W. (2025). Multi-dimensional anomaly detection and fault localization in microservice architectures: A dual-channel deep learning approach with causal inference for intelligent sensing. *Sensors*, 25(11), 3396.
- Xing, S., & Wang, Y. (2025). Proactive data placement in heterogeneous storage systems via predictive multi-objective reinforcement learning. *IEEE Access*, 13, 117986-117998.
- Ji, E., Wang, Y., Xing, S., & Jin, J. (2025). Hierarchical reinforcement learning for energy-efficient API traffic optimization in large-scale advertising systems. *IEEE Access*.
- Xing, S., & Wang, Y. (2025). Cross-modal attention networks for multi-modal anomaly detection in system software. *IEEE Open Journal of the Computer Society*.
- Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. *Sensors*, 24(24), 7949.
- Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. *Applied Sciences*, 15(2), 951.
- Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. *IEEE Access*, 12, 195974-195985.
- Hu, X., Zhao, X., Wang, J., & Yang, Y. (2025). Information-theoretic multi-scale geometric pre-training for enhanced molecular property prediction. *Plos one*, 20(10), e0332640.
- Wang, M., Zhang, X., Yang, Y., & Wang, J. (2025). Explainable machine learning in risk management: Balancing accuracy and interpretability. *Journal of Financial Risk Management*, 14(3), 185-198.
- Yang, J., Li, P., Cui, Y., Han, X., & Zhou, M. (2025). Multi-sensor temporal fusion transformer for stock performance prediction: An adaptive Sharpe ratio approach. *Sensors*, 25(3), 976.