Light but Sharp: SlimSTAD for Real-Time Action Detection from Sensor Data

Wei Cui[1], Lukai Fan[2], Zhenghua Chen[3], Min Wu[1], Shili Xiang[1], Haixia Wang[2], Bing Li[4]*

[1] Institute for Infocomm Research (I²R), Agency for Science, Technology and Research (A⋆STAR), Singapore
[2] Shandong University of Science and Technology, China
[3] James Watt School of Engineering, University of Glasgow, UK
[4] University of Electronic Science and Technology of China (UESTC), China
{cui_wei, xiang_shili, wu_min}@a-star.edu.sg, {hxwang, 202382080045}@sdust.edu.cn,
zhenghua.chen@glasgow.ac.uk, bing_li@uestc.edu.cn

Abstract

Sensory Temporal Action Detection (STAD) aims to localize and classify human actions within long, untrimmed sequences captured by non-visual sensors such as WiFi or inertial measurement units (IMUs). Unlike video-based TAD, STAD poses unique challenges due to the low-dimensional, noisy, and heterogeneous nature of sensory data, as well as the real-time and resource constraints on edge devices. While recent STAD models have improved detection performance, their high computational cost hampers practical deployment. In this paper, we propose SlimSTAD, a simple yet effective framework that achieves both high accuracy and low latency for STAD. SlimSTAD features a novel Decoupled Channel Modeling (DCM) encoder, which preserves modality-specific temporal features and enables efficient inter-channel aggregation via lightweight graph attention. An anchor-free cascade predictor then refines action boundaries and class predictions in a two-stage design without dense proposals. Experiments on two real-world datasets demonstrate that SlimSTAD outperforms strong video-derived and sensory baselines by 2.1 *m*AP@Avg, while significantly reducing GFLOPs, parameters, and latency, validating its effectiveness for real-world, edge-aware STAD deployment.

Code — https://github.com/windofshadow/SlimSTAD

Introduction

Temporal Action Detection (TAD) aims to automatically identify human actions (e.g., walking, running, and falling) and their timing within long, untrimmed time series. TAD originates from the computer vision community due to growing demands for video surveillance (Vahdani and Tian 2022). However, the ability of vision-based TAD is often limited in low-light or occluded environments, as line-of-sight and illumination are strict prerequisites for cameras. Moreover, the use of cameras also raise serious privacy concerns, especially in intimate or sensitive spaces like restrooms, bedrooms, or healthcare settings (Sun et al. 2022).

With the increasing prevalence of low-cost, off-the-shelf IoT sensing devices, such as WiFi devices, smartphone IMUs, or RF sensors, there is increasing interest in passive, long-term monitoring using continuous data streams from sensors. Unlike cameras, sensors are often more resilient to working conditions. For example, WiFi allows Non-Line-of-Sight (NLOS) monitoring by sensing variations in signal propagation (i.e., multi-path effect) caused by human motion, even when the subject is not directly visible to the transmitter or receiver (Yang et al. 2023). Additionally, sensors do not capture biometric or visual information, offering a more privacy-conscious approach. Sensory Temporal Action Detection (STAD) is particularly valuable for real-world applications like healthcare monitoring, smart home (Yang et al. 2018), and embodied intelligence (Roy et al. 2021).

Recently, there has been impressive progress in deep neural methods for STAD, ranging from adaptations of video-based models to dedicated models for sensory modalities. However, real-world deployment remains far from satisfactory. In particular, STAD faces unique challenges stemming from the low-dimensional, noisy, and heterogeneous nature of sensory signals. Unlike video data, where the spatial-temporal features can be well handled by modern pre-trained backbones (e.g., I3D (Carreira and Zisserman 2017), DETR (Zhu et al. 2020), etc.), sensory modalities often lack spatial structure, exhibit modality-specific temporal dynamics, and suffer from low signal-to-noise ratios. As a result,

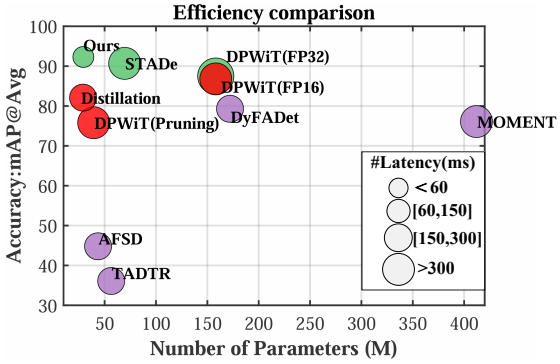


Figure 1: *m*AP-efficiency trade-off on the Sensor dataset. Our model is of the best balance of accuracy, size, and latency, compared with video-based TAD (purple), dedicated STAD (green), and lightweighting models (red).

*Corresponding Author

directly applying video-based TAD models to sensory data often leads to substantial performance declines. As shown in Fig. 1, state-of-the-art (SOTA) video TAD models TADTR and DyFADet underperform dedicated models by over 20 mAP points.

Moreover, STAD is typically intended for edge-based, long-term monitoring, where data acquisition, preprocessing, and inference have to be performed locally to ensure real-time responsiveness. This imposes strict constraints on latency and computational efficiency. However, as shown in Fig. 1, SOTA models remain heavyweight and computationally intensive, limiting their practicality for time-sensitive deployment. For example, even on a high-end edge-AI device like the NVIDIA Jetson Orin, processing a WiFi CSI record using DPWiT (Liu et al. 2025) takes 1.66 seconds, which fails to meet real-time requirements for human activity monitoring (Stisen et al. 2015; Lane et al. 2016) (i.e., strides $\leq$ 1Hz). Although model compression techniques such as pruning (Han et al. 2015), quantization (Jacob et al. 2018), or knowledge distillation (Hinton, Vinyals, and Dean 2015) have been proposed to improve efficiency, they often come at the cost of reduced detection accuracy. As illustrated in Fig. 1, none of the lightweight variants is able to maintain accuracy comparable to full-sized models.

To address challenges of low-quality sensory signals and resource constraints on edge deployment highlighted above, we propose a simple yet effective model, SlimSTAD, that improves both efficiency and effectiveness for temporal action detection on sensory data. Specifically, SlimSTAD introduces a structure-aware STAD framework composed of a lightweight feature encoder and an anchor-free cascade predictor. The encoder first applies per-channel temporal convolutions to preserve intra-channel locality, then leverages a lightweight graph-attention mechanism to aggregate inter-channel and temporal dependencies. This design naturally aligns with the inherent heterogeneity of sensory modalities and produces compact yet informative representations. Furthermore, to reduce complexity while maintaining accuracy, SlimSTAD employs a two-stage anchor-free prediction head. The first stage estimates coarse action boundaries and categories at each time step, while the second stage refines them via boundary-aware pooling and soft category fusion, enabling precise localization and robust classification, without relying on dense proposals or anchors. Remarkably, the overall design remains extremely simple yet powerful, significantly reducing model size while improving accuracy compared to heavyweighted counterparts (as shown in Fig. 1).

Experiments on two real-world datasets against strong video-based and dedicated STAD baselines demonstrate the strong performance of SlimSTAD. It improves mAP@Avg by 2.1 points over the best value of baselines, while reducing GFLOPs by 47.3%, parameter count by 48.8%, and inference latency by 27.9%. These results underscore SlimSTAD's effectiveness and efficiency for real-world, resource-constrained STAD applications.

Related Work

Video-based Temporal Action Detection

Temporal Action Detection (TAD) has been extensively studied in computer vision, primarily using video inputs. Deep learning-based TAD models typically fall into two categories: two-stage models (e.g., BSN (Lin et al. 2018), BMN (Lin et al. 2019), and GTAN (Long et al. 2019)), which separate proposal localization and classification, and one-stage models (e.g., AFSD (Lin et al. 2021) and SSTAD (Buch et al. 2019)), which perform both simultaneously end-to-end. Recent advances in Transformer-based models, e.g., TALLFormer (Cheng and Bertasius 2022) and TADTR (Liu et al. 2022), improves localization quality by capturing long-range dependencies. However, these models rely on the rich spatial-temporal structure of visual data and heavy pretrained backbones (e.g., I3D, DETR). Directly applying them to low-dimensional, noisy sensory data is suboptimal (Liu et al. 2025), leading to significant performance declines for STAD.

Sensory Temporal Action Detection

STAD poses unique challenges from the low-dimensional, noisy, and heterogeneous nature of sensor data. While earlier work in sensory activity recognition (Wang et al. 2023; Tian et al. 2018) has shown promising progress, detecting multiple actions from long, untrimmed sequences remains challenging. Recently, dedicated models, e.g., DPWiT (Liu et al. 2025) and STADe (Li et al. 2025), achieve strong STAD performance. DPWiT fuses global frequency-aware and local context via a Cross-Transformer. STADe builds upon an I3D backbone combining with Fourier-based kernels to model temporal-frequency patterns. However, their heavy computational cost makes them impractical for real-time edge deployment, highlighting the need for lightweight STAD architectures.

In contrast, our proposed model achieves both lightweight design and high effectiveness. It does this by explicitly modeling channel-wise temporal dynamics and inter-channel dependencies through a decoupled encoder, further enhanced by an anchor-free cascade predictor, ensuring optimal efficiency and accuracy for real-time, on-device deployment.

Method

Problem Definition

Sensory Temporal Action Detection (STAD) targets automatically localize and identify actions, within untrimmed, multivariate sensory time series. Given a dataset $\mathcal{D} = \{d_1, d_2, ..., d_n\}$, where each record $d_i = \{X, \Psi\}$ comprises a sensory sequence $X \in \mathbb{R}^{C \times T}$ with C channels and T time steps, and a corresponding annotation set $\Psi = \{(\phi_m, y_m)_{m=1}^{M}\}$, the goal is to detect all M action instances within X. Each instance is denoted by a tuple (ϕ_m, y_m), where $\phi_m = (\psi_m, \xi_m)$ marks the start and end times, and $y_m \in \{1, ..., K\}$ indicates the action category.

Model Overview

SlimSTAD (framework shown in Fig. 2) takes an untrimmed sensory time series as input. The data first passes through a

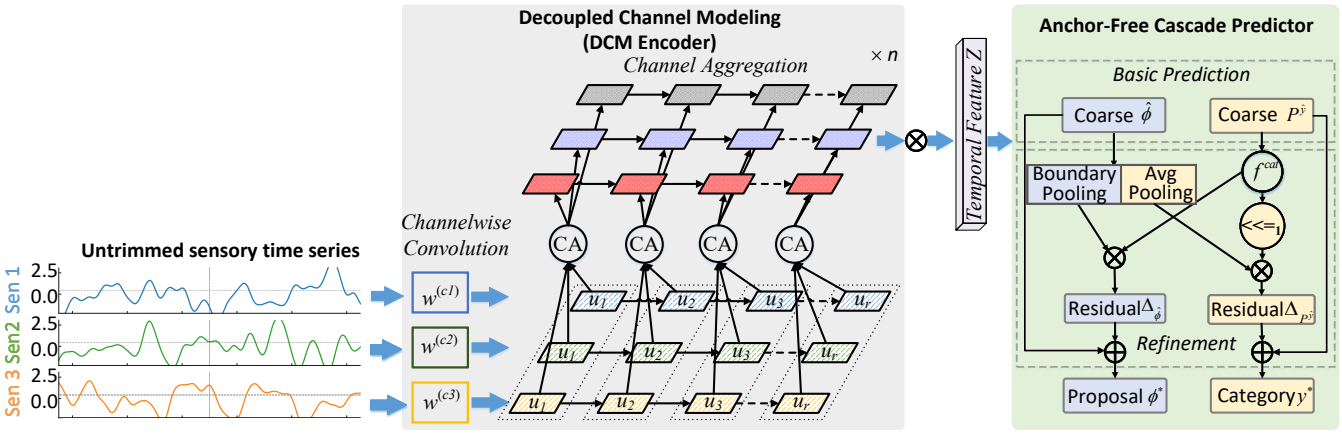


Figure 2: The framework of our SLIMSTAD model. The DCM encoder extracts discriminative features via channelwise convolution and graph-based channel aggregation. An anchor-free cascade predictor then performs coarse-to-fine action detection, refining boundaries and categories using boundary-aware pooling and historical category context.

Decoupled Channel Modeling (DCM) encoder to extract discriminative features. DCM begins with *channelwise temporal convolutions*, which independently capture intra-channel patterns using separate 1D kernels. This is followed by a *channel aggregation (CA)* module that constructs a spatiotemporal graph over the feature map and applies graph attention to model inter-channel dependencies. The resulting features are then fed into an *Anchor-Free Cascade Predictor* for action prediction. This predictor operates in two stages: the first stage performs anchor-free predictions to estimate coarse action boundaries and category logits at each time step; the second stage refines these predictions via boundary-aware pooling, soft category embeddings, and historical category cues. Finally, a lightweight CNN predicts residual offsets and refines coarse predictions. We elaborate on each component in the following sections.

Decoupled Channel Modeling

Existing STAD models often *blindly inherit video backbones*, such as I3D and R3D, as their feature extractors. Unlike visual data where spatial-temporal features are often entangled and benefit from shared kernels, sensory data channels are inherently *heterogeneous and semantically distinct*. For instance, smartphones collect signals from accelerometers, gyroscopes, and gravity sensors, while WiFi CSI channels are different subcarriers via OFDM. These channels are heterogeneous in semantics yet temporally aligned and often exhibit complementary correlations[1] in response to human motion. Traditional approaches either prematurely fuse or treat channels in isolation, losing modality-specific nuances or incurring redundancy.

To directly address this fundamental challenge, we propose a lightweight yet expressive Decoupled Channel Modeling (DCM) encoder. DCM first preserves modality-specific temporal dynamics via independent channel-wise convolutions, then efficiently models complex, non-linear inter-channel dependencies through a lightweight graph attention mechanism

[1] As illustrated in Fig.3, WiFi CSI channels show complex inter-channel correlations.

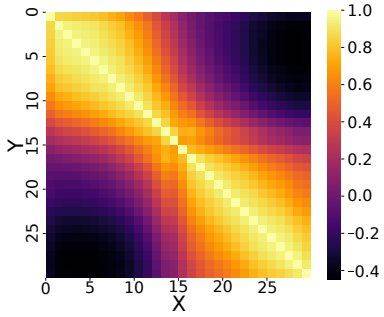


Figure 3: An example of Pearson coefficient matrix of channels/sub-carriers of WiFi CSI data.

applied within local temporal chunks. This decoupling and re-aggregation strategy is crucial for extracting informative signals from noisy, low-dimensional, and heterogeneous sensory data, without imposing vision-centric spatial assumptions. DCM consists of the following two core components:

i) Channelwise Convolution. We employ channelwise temporal convolution to enable independent intra-channel feature extraction with low computational cost. Given the input sensory sequence $X \in \mathbb{R}^{C \times T}$, a channelwise temporal convolution independently applies 1D convolution along the temporal dimension of each channel:

$$\widetilde{x}_t^{(c)} = \sum_{i=0}^{k-1} w_i^{(c)} \cdot x_{t \cdot s+i}^{(c)}, \quad \forall t \in \{0, \ldots, T'-1\}, \qquad (1)$$

where $x^{(c)} \in \mathbb{R}^T$ is the temporal signal from c-th channel and $w^{(c)}$ is its channel-specific kernel. $\widetilde{x}^{(c)} \in \mathbb{R}^{T'}$ is the output feature, with $T' = \lfloor \frac{T-k}{s} \rfloor + 1$, s is the stride. Unlike shared convolutions in CNNs, each channel keeps its own learnable kernel for preserving modality-specific features.

By applying the channel-wise convolution across all channels with d kernels, we obtain the feature map:

$$\widetilde{X} = \text{ChannelConv}(X) \in \mathbb{R}^{C \times T' \times d}. \qquad (2)$$

ii) Channel Aggregation. We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model inter-channel correlations and integrate information across channels. To preserve temporal resolution, we perform aggregation within a local temporal chunk rather than across the entire sequence. Specifically, we define a chunk u_i covering time steps $l \in \{i+1, \ldots, i+r\}$, and construct nodes $v_i \in \mathcal{V} = \{t_l^{(c)}\}$ for all channels $c \in \{1, \ldots, C\}$ and time steps l within the chunk. Each node is associated with a feature vector $\mathbf{h}_i \in \mathbb{R}^d$, extracted from the corresponding slice of $\widetilde{X}$. We then apply graph attention (Veličković et al. 2018) to update each node as:

$$\mathbf{z}_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W}\mathbf{h}_j\right), \tag{3}$$

where $\mathcal{N}(i)$ is the set of neighbors of node i (we set $\mathcal{N}(i) = \mathcal{V}$ by default), $\mathbf{W} \in \mathbb{R}^{d' \times d}$ is a learnable linear transformation, $\sigma(\cdot)$ is a nonlinearity (e.g., ReLU), and α_{ij} is the attention coefficient, computed as:

$$\alpha_{ij} = \frac{\exp\left(\sigma(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_j])\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\sigma(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_k])\right)}, \tag{4}$$

where $\mathbf{a} \in \mathbb{R}^{2d'}$ is a single-layer feedforward neural network, $\mathbf{W}$ is a learnable linear transformation, and $||$ denotes vector concatenation.

The final graph-enhanced feature matrix is:

$$Z = \mathrm{GCA}(\widetilde{X}) \in \mathbb{R}^{C \times T' \times d'} \tag{5}$$

Complexity. Channelwise convolution has complexity $O(C \cdot d \cdot k \cdot T)$, and channel aggregation has $O(C \cdot |\mathcal{N}| \cdot T \cdot d \cdot d')$. In contrast, conventional video backbones like I3D, have a much higher complexity of $O(L \cdot C^2 \cdot k^3 \cdot T \cdot H \cdot W)$[2], dominated by spatial operations. Moreover, since Memory Access Cost (MAC) is the primary efficiency bottleneck on edge devices (Sze et al. 2017), our design yields a MAC of $O(T \cdot C^2 \cdot d)$, far smaller than the $O(T \cdot C^2 \cdot H \cdot W)$ of I3D.

Notably, DCM's efficiency is not achieved through post-hoc compression or pruning, but stems from a design philosophy: sensory signals lack the spatial redundancy present in visual data. DCM offers a simpler yet more flexible and computationally efficient inductive bias for sensory data, by focusing on independent temporal processing per channel followed by a highly adaptable global aggregation

Anchor-Free Cascade Predictor

To decode the representations from DCM, we introduce an effective two-stage anchor-free prediction module consisting of a basic predictor and a refinement stage.

Basic Prediction We first adopt a dense anchor-free strategy to generate initial action proposals at each time step. Given the encoded feature sequence Z, we apply separate temporal convolutional heads to obtain coarse estimates of action boundaries ϕ and category logits y. Formally, for each time step i, the outputs are computed as:

$$\hat{\phi}_i = \mathrm{Linear}_{\mathrm{loc}}(\mathrm{Conv}(Z)), \quad P_i^{\hat{y}} = \mathrm{Linear}_{\mathrm{cls}}(\mathrm{Conv}(Z)). \tag{6}$$

This design avoids predefined anchors and enables flexible, dense predictions refined via NMS (Bodla et al. 2017).

Instead of regressing absolute boundaries (ψ_i, ξ_i), we predict relative offsets from the center time step i to improve training stability. The start and end positions are then computed as:

$$\hat{\psi}_i = i - \hat{d}_i^s, \quad \hat{\xi}_i = i + \hat{d}_i^e, \tag{7}$$

where $\hat{d}_i^s$ and $\hat{d}_i^e$ are distances to the start and end.

Refinement Action boundaries in sensor data are often ambiguous due to inherent signal noise and gradual transitions between activities. The single-stage predictors struggle with precise localization, while anchor-based methods are sensitive to predefined scales, which are inferior for detecting accurate action durations for sensory data. We propose a refinement stage to integrates three important contexts for robust boundary and category prediction: *i*) *boundary pooling* extracts salient features around coarse boundaries, making it robust to noisy initial estimates; *ii*) *soft category embeddings* that inject category-guided action duration priors (e.g., *sit* tends to be short, while *walk* is longer); and *iii*) *historical category* cues (via left-shifting) which explicitly model temporal dependencies between actions (e.g., *sitting* precedes *standing*).

The *boundary pooling* extracts salient features around the predicted start and end points $(\hat{\psi}_i, \hat{\xi}_i)$ of each proposal:

$$\begin{aligned} \tilde{f}_{\mathrm{start}}(i,k) &= \max_{j \in \left[\hat{\psi}_k - \frac{w_k}{\epsilon_a},\ \hat{\psi}_k + \frac{w_k}{\epsilon_a}\right]} Z(i,j), \quad i = 1, \ldots, C, \\ \tilde{f}_{\mathrm{end}}(i,k) &= \max_{j \in \left[\hat{\xi}_k - \frac{w_k}{\epsilon_b},\ \hat{\xi}_k + \frac{w_k}{\epsilon_b}\right]} Z(i,j), \quad i = 1, \ldots, C, \end{aligned} \tag{8}$$

where $w_k = \xi_k - \psi_k$ is the proposal length, and ϵ_a and ϵ_b control external/internal span around the boundaries.

Meanwhile, action categories often imply useful priors for action duration. To inject such priors, *soft category embedding* is generated via $\tilde{f}_i^{\mathrm{cat}} = P_i^{\hat{y}} W^{\mathrm{cat}}$, where $W^{\mathrm{cat}} \in \mathbb{R}^{K \times d}$ is a learnable matrix. The concatenated feature $[\tilde{f}_i^{\mathrm{start}}; \tilde{f}_i^{\mathrm{end}}; \tilde{f}_i^{\mathrm{cat}}]$ is passed through a CNN to produce a residual correction to the proposal:

$$\Delta_{\hat{\phi}_i} = \mathrm{CNN}([\tilde{f}_i^{\mathrm{start}}; \tilde{f}_i^{\mathrm{end}}; \tilde{f}_i^{\mathrm{cat}}]). \tag{9}$$

Beyond proposal refinement, we also enhance category prediction using both proposal-aware and historical context. We first refine category prediction by summarizing the features within each proposal:

$$\tilde{f}^{\mathrm{prop}} = \mathrm{AvgPooling}(\mathrm{Conv}(Z[\hat{\psi}_i : \hat{\xi}_i])), \tag{10}$$

where $Z[\hat{\psi}_i : \hat{\xi}_i]$ denotes the temporal slice of the encoded features within the initially predicted boundaries.

To capture temporal priors (e.g., *sitting* often precedes *standing*), we generate a historical embedding by left-shifting[3] the category embedding: $\tilde{f}_i^{\mathrm{his}} <<=_1 (\tilde{f}_i^{\mathrm{cat}})$.

[2]Even when sensory signals are reshaped to $1 \times T \times \{H \times W\} = C$ for compatibility, the resulting complexity remains at $O(d \cdot k^3 \cdot T \cdot C)$, still higher than ours.

[3]Left-truncation and zero-padding are used for alignment.

Finally, we concatenate the contextual feature $\tilde{f}_i^{\text{prop}}$ and historical feature $\tilde{f}_i^{\text{his}}$, and feed them into a CNN to predict the final category:

$$\Delta_{P_i^{\hat{y}}} = \text{CNN}([\tilde{f}_i^{\text{prop}}; \tilde{f}_i^{\text{his}}]). \quad (11)$$

This refinement handles the continuous and uncertain nature of sensor-based action well. Furthermore, unlike video-based TAD models that often refine thousands of dense proposals, our refinement operates on high-confidence coarse proposals provided by basic prediction. This sparse-to-dense refinement strategy, combined with lightweight CNNs, ensures substantial performance gains with minimal overhead[4], making it practical for edge deployment.

Training and Inference

Training During training, a temporal index i is labeled as positive if it falls within a ground-truth proposal $\phi_j = (\psi_j, \xi_j)$, i.e., $\psi_j \le i \le \xi_j$. To ensure high-quality supervision, the refinement predictor P_ϕ^c is only applied to positive proposals whose predicted coarse segment $\hat{\phi}_j = (\hat{\psi}_j, \hat{\xi}_j)$ achieves an Intersection-over-Union (IoU) greater than 0.5 with the corresponding ground-truth ϕ_j.

The overall training objective combines both stages (coarse and refined) for boundary and category predictions:

$$\mathcal{L} = \alpha \mathcal{L}_\phi^b + \mathcal{L}_y^b + \alpha \mathcal{L}_\phi^c + \mathcal{L}_y^c, \quad (12)$$

where α is a balance weight for the two stages. $\mathcal{L}_\phi^b = \frac{1}{N}\sum_i \mathbb{I}(y_i \ge 1)\left(1 - \frac{|\hat{\phi}_i \cap \phi_i|}{|\hat{\phi}_i \cup \phi_i|}\right)$ is the IoU loss for coarse boundaries. $\mathcal{L}_\phi^c = \frac{1}{N_c}\sum_i \mathbb{I}(y_i \ge 1)\left|\hat{\Delta}_i - \Delta_i\right|$ is the L1 loss between predicted offsets and ground-truth residuals for refinement. $\mathcal{L}_y^b$ and $\mathcal{L}_y^c$ are focal losses (Lin et al. 2017) computed on the predicted category logits from the coarse and refined stages, respectively.

Inference During inference, we fuse the initial and refined predictions to obtain the final estimates:

$$\psi_i^\star = \frac{1}{2}(\hat{\phi}_i + \Delta_{\hat{\phi}_i}), \quad y_i^\star = \arg\max_j (P_i^{\hat{y}}(j) + \Delta_{P_i^{\hat{y}}}(j)) \quad (13)$$

Finally, we apply NMS (Bodla et al. 2017) over $(\psi_i^\star, y_i^\star)$ to eliminate redundant predictions and retain the most confident action segments.

Experiments

Experimental Setups

Datasets We adopt two real-world datasets with different types of sensors (Table 1): a Wi-Fi Channel State Information (CSI) sensing dataset and a smartphone sensor dataset.

Wi-Fi HAR Dataset (Liu et al. 2025). This WiFi-CSI temporal action detection dataset was collected in an empty office (7 m × 12 m × 2.5 m) using two Intel 5300-NIC laptops (one antenna, 30 subcarriers, 500 Hz). Three volunteers performed seven activities: walk, run, jump, wave, fall, sit, stand—between the transmitter (TX) and receiver (RX). The dataset contains 553 untrimmed recordings and 2,114 annotated action instances, split 70:30 for training/testing. Instance counts and duration statistics are shown in Table 4.

[4]Refinement attains a 3.5-point improvement in mAP while adding only 10% additional cost.

Property	*Wi-Fi HAR*	*Sensor*
Sampling Rate (Hz)	500	200
# Time Series	553	3,615
Avg. Length	30,000	12,000
# Act. Categories	7	7
# Avg. Act. per TS	3.82	4.31
Sensor Type	WiFi CSI	Accelerometer, gravity sensor, and gyroscope (@Smartphone)
Actions List	walk, run, jump, wave, fall, sit, and stand	walk, run, stand, upstairs, downstairs, lie, and sit

Table 1: Dataset Statistics. Summary of the two datasets used in our experiments.

Sensor Dataset (Li et al. 2025). This dataset was collected using an Android app that recorded three inertial sensors of smartphones: tri-axial acceleration, gravity, and gyroscope, at 200 Hz. Six volunteers participated. Each one-minute trial involved holding the phone in hand and randomly performing 4–6 actions from: walking, running, standing, upstairs, downstairs, lying, and sitting, yielding a roughly balanced class distribution. The final dataset contains 3,615 untrimmed sequences and 15,590 annotated action instances, each labeled with start time, end time, and class. We adopt a 70:30 train–test split. Instance counts and duration statistics appear in Table 5.

Each dataset is evaluated end-to-end without window segmentation. Full untrimmed recordings (∼30s for WiFi-HAR and ∼60s for Sensor) are fed to the model, ensuring natural transitions and consistent train/test splits. For both datasets, we adopt a 70:30 train/test split over full recordings.

Baselines We compare against state-of-the-art (SOTA) models from both video-based and signal-specific STAD approaches: *Video-based TAD models*: (1) AFSD (Lin et al. 2021): Anchor-free model with boundary-aware features. (2) TADTR (Liu et al. 2022): Transformer-based model using deformable temporal attention. (3) DyFADet (Yang et al. 2024): Employs dynamic feature aggregation for adaptive receptive fields. *Dedicated STAD models*: (4) MOMENT (2024): A LLM-based model use T5-large as backbone and pretrained on a large-scale Time-series Pile corpus. (5) DPWiT (Liu et al. 2025): Combines dual feature pyramids with mask and cross attention to capture frequency and local variations. (6) STADe (Li et al. 2025): Leverages multi-scale temporal–spatial representations for signal-based action detection.

Metrics We follow (Liu et al. 2025) to report mean Average Precision (mAP) at temporal intersection-over-union (tIoU) thresholds from 0.3 to 0.7 (step 0.1). A prediction is correct if its tIoU with ground truth exceeds the threshold and the label matches. The overall mAP is averaged over all classes. We also measure GFLOPs, parameter count (M), and inference latency (ms) to evaluate model efficiency.

Type	Method	(a) *m*AP@tIoU (%)						(b) Model Complexity			
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**Avg ↑**	**GFLOPs ↓**	**Param (M)↓**	**Latency (GPU, ms)↓**	**Latency (EAI, ms)↓**
Video-based	AFSD (2021)	46.6	45.4	42.4	37.9	25.4	39.5	91.0	82.5	71.8	-
	TADTR (2022)	63.3	59.9	57.4	51.8	38.2	54.1	185.4	102.6	106.0	-
	DyFADet (2024)	66.8	64.2	62.4	56.4	40.1	58.0	304.0	183.2	75.7	-
Dedicated	DPWiT (2025)	85.5	83.0	77.3	72.1	54.5	74.5	44.1	150.2	333.3	1666.7
	STADe (2025)	84.9	84.2	82.7	**78.8**	54.5	77.1	158.3	75.9	267.6	1041.6
	SLIMSTAD (Ours)	**89.4** (+5.3%)	**88.1** (+4.6%)	**84.2** (+1.8%)	77.3 (–1.9%)	**59.7** (+9.5%)	**79.7** (+3.4%)	**31.8** (–27.9%)	**26.7** (–64.8%)	**58.8** (–18.1%)	**282.5** (–61.3%)

Table 2: Main Results: Comparison on the ***WiFi-HAR*** dataset. The best results are marked in **bold**, second best are underlined. Values in parentheses indicate relative improvement/reduction over the best baseline. Green denotes improvement, red denotes reduction.

Type	Method	(a) *m*AP@tIoU (%)						(b) Model Complexity			
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**Avg ↑**	**GFLOPs ↓**	**Param (M)↓**	**Latency (GPU, ms)↓**	**Latency (EAI, ms)↓**
Video-based	AFSD (2021)	53.7	49.9	45.0	40.3	34.9	44.8	45.7	43.6	129.6	-
	TADTR (2022)	45.4	40.5	35.6	31.4	27.4	36.1	76.3	56.4	77.1	-
	DyFADet (2024)	87.3	85.2	81.0	74.7	68.0	79.3	283.9	172.1	112.2	-
Dedicated	MOMENT (2024)	82.3	80.2	76.8	73.3	68.0	76.1	126.0	412.1	185.4	-
	DPWiT (2025)	92.8	92.0	89.6	84.1	79.1	87.5	77.3	158.2	303.9	1527.3
	STADe (2025)	94.0	93.2	91.7	89.5	85.3	90.7	109.9	69.5	168.7	729.6
	SLIMSTAD (Ours)	**96.4** (+2.6%)	**95.6** (+2.6%)	**93.6** (+2.1%)	**90.4** (+1.0%)	**85.5** (+0.2%)	**92.3** (+1.8%)	**15.2** (–66.7%)	**29.3** (–32.8%)	**48.0** (–37.7%)	**187.3** (–74.3%)

Table 3: Main Results: Comparison on the ***Sensor*** dataset.

Category	Walk	Run	Jump	Wave	Fall	Sit	Stand
# Instances	394	361	347	335	332	225	120
Avg. Duration (s)	16	17	13	18	13	13	11
Max. Duration (s)	30	25	20	30	25	40	20
Min. Duration (s)	10	5	5	10	5	5	5

Table 4: WiFi-HAR Dataset Statistics. Instance counts and duration statistics for each activity.

Category	Walk	Sit	Stand	Lie	Ascend Stairs	Descend Stairs	Run
# Instances	3059	2956	2361	1195	1627	1556	1829
Avg. Duration (s)	15	10	12	13	12	12	14
Max. Duration (s)	56	64	46	46	51	43	56
Min. Duration (s)	3	–	5	10	5	5	5

Table 5: Sensor Dataset Statistics. Instance counts and duration statistics for each activity.

Implementation Details We used Python 3.8 and PyTorch 1.10 on an AMD EPYC 7642 CPU and RTX 3090 GPU (Ubuntu). Edge latency was tested on Jetson Orin Nano Super 8GB (JetPack 6.1, PyTorch 2.5). The model was trained with Adam. For WiFi-HAR: LR=2×10^{-4}, weight decay=1×10^{-4}, batch=2, epochs=80, loss factor=6. For Sensor: LR=1×10^{-3}, weight decay=1×10^{-3}, batch=4, epochs=60, loss factor=10. Both used 4 DCM layers (4 heads each), with batch norm, pooling, and dropout (keep=0.5). Channel dims: 64, 256, 512, 832 (WiFi-HAR), 64, 128, 832, 1024 (Sensor). Soft-NMS (Bodla et al. 2017) was applied (sigma=0.5, threshold=0.1).

Main Results

Table 2 and Table 3 report the *m*AP scores under varying tIoU thresholds (0.3–0.7), and model complexity.

Firstly, our model SLIMSTAD consistently achieves the *best mAP for almost all IoU settings*. On WiFi-HAR dataset, our method has an *m*AP@Avg of 79.7%, outperforming the SOTA dedicated models STADe and DPWiT with a margin of 2.6 and 5.2 points, respectively. Further, the three video TAD models, DyFADet, TADTR, and AFSD show inferior performance compared with ours (more than 20 points margin). This indicates a fundamental mismatch between the characteristics of video data and sensory signals, suggesting direct application of video-based models is suboptimal. The trend on Sensor dataset is similar, our model again achieves the best *m*AP@Avg of 92.3%, outperforms STADe (90.7%) and DPWiT (87.5%). DyFADet achieves only 79.3%, and other video models fall below 45%.

Secondly, our model also has *outstanding efficiency*. On both datasets, it achieves the lowest GFLOPs of 31.8 on WiFi-HAR and 15.2 on Sensor, which is up to 9.6× smaller than video-based model DyFADet (304.0 and 283.9 GFLOPs) and 5× smaller than dedicated model STADe (158.3 and 109.9 GFLOPs). Furthermore, our model also have the smallest

model size, with only 26.7M and 29.3M parameters, which is 5.9× smaller than DPWiT (158.2M) and less than half the size of STADe (75.9M and 69.5M). This shows a superior model complexity of our model.

Thirdly, our model achieves the *lowest inference latency* across both GPU and edge-AI (EAI) devices. Specifically, SlimSTAD obtains inference latencies of 58.8 ms (GPU) and 282.5 ms (EAI) on WiFi-HAR, and 48.0 ms (GPU) and 187.3 ms (EAI) on the Sensor dataset. Compared with DPWiT and STADe, our method achieves up to 5.7× and 8.3× lower latency on GPU and EAI, respectively. Notably, the speed-up on EAI devices is more obvious, as SlimSTAD significantly reduces MAC, a major bottleneck on edge hardware, by avoiding spatial convolutions used in video-centric backbones (e.g., I3D in STADe). This highlights our model's suitability for real-time deployment, especially in latency-sensitive or resource-constrained edge scenarios.

The above results demonstrates the effectiveness of our lightweight design: by integrating an efficient DCM encoder and a cascade predictor, our model effectively captures informative patterns in sensory data. This architecture yields substantial improvements in both precision and computational efficiency for sensory-based TAD tasks.

Model	*m*AP@tIoU				Complexity	
	0.3	0.5	0.7	Avg ↑	GFLOPs ↓	Param (M) ↓
Backbone Replacement						
C3D	93.3	90.2	80.8	88.5	27.5	38.2
I3D	94.6	91.5	82.2	90.1	41.9	43.4
R(2+1)D	93.0	89.2	80.3	84.1	22.2	34.9
R3D	95.4	92.0	83.8	90.8	62.0	73.2
Design of DCM						
w/o Chann. Aggre.	95.0	90.9	81.3	89.7	15.1	28.3
Decoder						
w/o Refinement	94.7	91.1	78.9	88.8	13.8	24.2
Proposed Model						
SlimSTAD (Ours)	96.4	93.6	85.5	92.3	15.2	29.3

Table 6: Ablation Study: Performance under various configurations.

Ablation Study

To evaluate effectiveness of each components, we conduct an ablation study on the *Sensor* dataset, reporting *m*AP at tIoU thresholds 0.3, 0.5, and 0.7 (Table 6).

For backbone comparison, we evaluate four standard 3D CNNs: C3D (Tran et al. 2015), I3D (Carreira and Zisserman 2017), R(2+1)D (Tran et al. 2018), and R3D (Hara, Kataoka, and Satoh 2017). I3D and R3D perform competitively (90.1% v.s. 90.8%), while R(2+1)D performs the worst (84.1% *m*AP) due to its decoupled spatial-temporal convolutions limits modeling the spatiotemporal dependencies. Our model outperforms all with 92.3% *m*AP, owing to DCM's effective integration of localized temporal encoding and channel dependencies via graph attention. Removing graph-based channel aggregation (by replacing it a trivial linear layer) leads to a roughly 3 points performance drop, confirming the importance of integrating information from different sensor channel. Finally, the ablation in the refinement stage of the predictor (only keeps base predictor) leads to a 3.5 points performance drop, highlighting the importance of a refined prediction head. In conclusion, these results validate the complementary roles of DCM and our cascade decoder in achieving state-of-the-art performance.

Method	***m*AP ↑**	**Mem (MiB) ↓**	**Lat. (ms) ↓**	**Param (M) ↓**
DPWiT	87.5	3644	303.9	158.2
DPWiT_FP16	86.8	2926	278.2	158.2
DPWiT_Distilled	82.1	–	151.9	29.0
DPWiT_Pruned	75.8	–	233.6	39.3
SlimSTAD (Ours)	**92.3**	2471	48.0	29.3

Table 7: Comparison with conventional lightweighting strategies on the Sensor dataset.

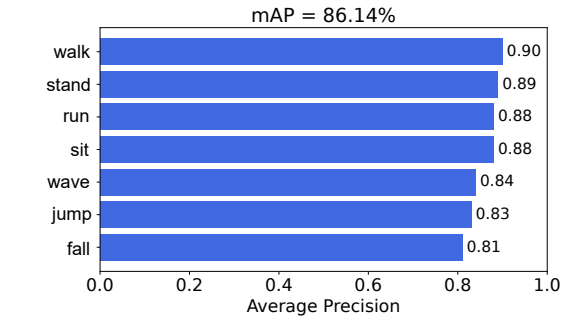
Comparison with Lightweighting Techniques

Table 7 compares SlimSTAD against several conventional lightweighting techniques, including quantization (Jacob et al. 2018), knowledge distillation (Hinton, Vinyals, and Dean 2015), and pruning (Han et al. 2015), on Sensor dataset. We select DPWiT as the base model and evaluate its FP16 quantized (DPWiT_FP16), distilled (DPWiT_Distilled), and channel-pruned (DPWiT_Pruned) variants. DPWiT_FP16 slightly reduces memory (19.7%) and latency (8.5%), but at the cost of accuracy (-0.7 mAP). DPWiT_Distilled achieves comparable size (29M) but incurs significantly higher latency (3.16×) and a notable accuracy drop (-5.5 mAP). DPWiT_Pruned reduces parameters (39.3M) yet sharply decreases accuracy (75.8 mAP) and remains high latency (233.6ms). In contrast, SlimSTAD achieves the highest *m*AP of 92.3 with only 48 ms latency. It provides the best trade-off between accuracy and efficiency, outperforming all three lightweighting techniques.

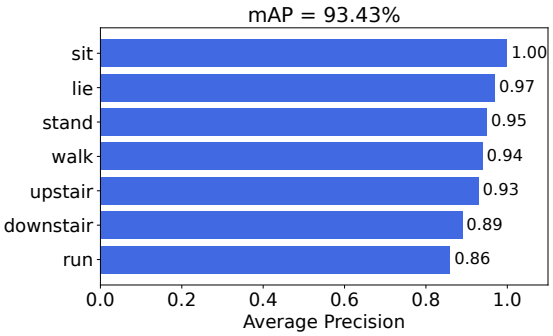
Model	mAP@Avg (%) ↑	GFLOPs ↓
TemporalMaxer (2023)	56.4	189.6
TriDet (2023)	55.0	239.5
SlimSTAD (Ours)	**92.3**	**31.8**

Table 8: Comparison with lightweight models.

We further compare SlimSTAD with two lightweight anchor-free models: TemporalMaxer (Tang, Kim, and Sohn 2023) and TriDet (Shi et al. 2023), which have been validated in recent large-scale benchmarks XRF V2 (Lan et al. 2025), for their outstanding efficiency. As shown in Table 8, SlimSTAD achieves +24.7% higher mAP@Avg (92.3 vs. 56.4/55.0) while requiring 6× fewer GFLOPs (31.8 vs. 189.6/239.5), highlighting its superior accuracy–efficiency trade-off.



(a) WiFi-HAR@tIoU=0.5



(b) Sensor@tIoU=0.5

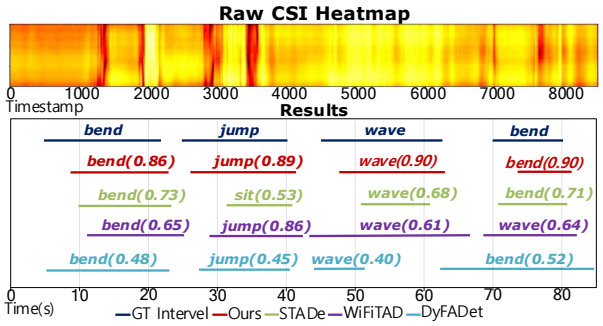
Figure 4: Categorical analyses on the two datasets. We set tIoU=0.5.

Categorical Performance

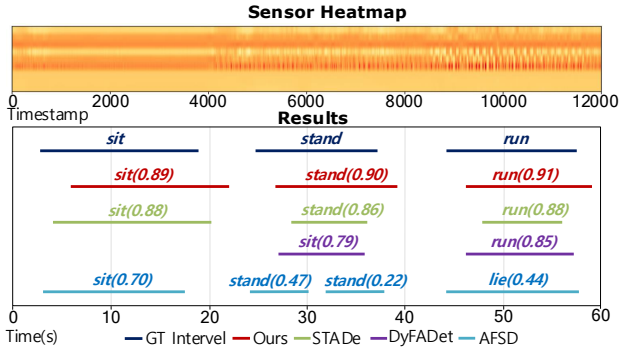
Fig. 4 shows the categorical performance of SLIMSTAD on both WiFi-HAR and Sensor datasets. The model maintains high precision across all activities, demonstrating robust generalization to different sensing modalities. On the Sensor dataset, performance is uniformly strong (mAP=93.43%), with high accuracy for stable actions such as *sit*, *lie*, and *stand*, and slightly lower scores for *run* (0.86) and *downstair* (0.89) due to faster dynamics. In contrast, WiFi-HAR poses greater challenges given its noisy CSI signals; yet the model achieves an average precision of ∼0.86, with most actions ranging from 0.81–0.90. Slightly reduced precision for *wave*, *jump*, and *fall* arises from their subtle motion transitions, highlighting the difficulty of coarse-grained CSI representations. Overall, these results confirm the robustness of SLIMSTAD across diverse sensor domains.

Visual Analysis

Fig. 5 presents the localization visualizations on both the WiFi-HAR and Sensor datasets, showing the ground-truth activity intervals alongside the best predicted intervals from four representative models. As illustrated, our method accurately detects both the candidate action intervals and their corresponding action type. In contrast, while STADe successfully identifies the correct actions, its predicted intervals show noticeable temporal shifts. DPWiT and DyFADet



(a) WiFi-HAR@tIoU=0.5



(b) Sensor@tIoU=0.5

Figure 5: Visualization: Spectral maps are obtained by averaging sensory channel values from examples in the datasets. Ground-truth and predicted proposals are marked by black and orange vertical lines, respectively. tIoU values and correct categories are indicated.

exhibit misclassifications on the WiFi-HAR dataset, with DyFADet also missing true action segments in the Sensor dataset. AFSD produces scattered predictions that fail to form coherent detection results. Overall, SLIMSTAD demonstrates superior temporal localization and category discrimination across both visualized datasets.

Conclusion

We proposed SLIMSTAD, a lightweight yet powerful model for sensory temporal action detection. By explicitly decoupling intra-channel temporal modeling and inter-channel dependency reasoning, combined with an anchor-free cascade predictor, SLIMSTAD significantly excels existing video-based and sensory-specific baselines in both effectiveness and efficiency on the WiFi-HAR and Sensor datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant no. 62476053, 62472262, and 62273213), Green Buildings Innovation Cluster 2.0 Challenge Call for Decarbonisation (GBIC R&D/2025/4), Fundamental Research Funds for the Central Universities under Grant ZYGX2024J003, and the Taishan Scholarship Construction Engineering.

References

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-NMS–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, 5561–5569.

Buch, S.; Escorcia, V.; Ghanem, B.; Fei-Fei, L.; and Niebles, J. C. 2019. End-to-end, single-stream temporal action detection in untrimmed videos. In *Procedings of the British Machine Vision Conference*. British Machine Vision Association.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Cheng, F.; and Bertasius, G. 2022. Tallformer: Temporal action localization with a long-memory transformer. In *European Conference on Computer Vision*, 503–521. Springer.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning*, 16115–16152. PMLR.

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Hara, K.; Kataoka, H.; and Satoh, Y. 2017. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, 3154–3160.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; and Kalenichenko, D. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2704–2713.

Lan, B.; Li, P.; Yin, J.; Song, Y.; Wang, G.; Ding, H.; Han, J.; and Wang, F. 2025. Xrf v2: A dataset for action summarization with wi-fi signals, and imus in phones, watches, earbuds, and glasses. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3): 1–41.

Lane, N. D.; Bhattacharya, S.; Georgiev, P.; Forlivesi, C.; Jiao, L.; Qendro, L.; and Kawsar, F. 2016. Deepx: A software accelerator for low-power deep learning inference on mobile devices. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 1–12. IEEE.

Li, B.; Duan, H.; Liu, Y.; Zhang, L.; Cui, W.; and Zhou, J. T. 2025. STADe: Sensory Temporal Action Detection via Temporal-Spectral Representation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9): 8117–8133.

Lin, C.; Xu, C.; Luo, D.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Fu, Y. 2021. Learning Salient Boundary Feature for Anchor-free Temporal Action Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3320–3329.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. BMN: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3889–3898.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision*, 3–19.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, X.; Wang, Q.; Hu, Y.; Tang, X.; Zhang, S.; Bai, S.; and Bai, X. 2022. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31: 5427–5441.

Liu, Z.; Zhang, L.; Li, B.; Zhou, Y.; Chen, Z.; and Zhu, C. 2025. WiFi CSI Based Temporal Activity Detection via Dual Pyramid Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 550–558.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 344–353.

Roy, N.; Posner, I.; Barfoot, T.; Beaudoin, P.; Bengio, Y.; Bohg, J.; Brock, O.; Depatie, I.; Fox, D.; Koditschek, D.; et al. 2021. From machine learning to robotics: Challenges and opportunities for embodied intelligence. *arXiv preprint arXiv:2110.15245*.

Shi, D.; Zhong, Y.; Cao, Q.; Ma, L.; Li, J.; and Tao, D. 2023. Tridet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18857–18866.

Stisen, A.; Blunck, H.; Bhattacharya, S.; Prentow, T. S.; Kjærgaard, M. B.; Dey, A.; Sonne, T.; and Jensen, M. M. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 127–140.

Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; and Liu, J. 2022. Human action recognition from various data modalities: A review. *IEEE transactions on pattern analysis and machine intelligence*, 45(3): 3200–3225.

Sze, V.; Chen, Y.-H.; Yang, T.-J.; and Emer, J. S. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12): 2295–2329.

Tang, T. N.; Kim, K.; and Sohn, K. 2023. Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization. *arXiv preprint arXiv:2303.09055*.

Tian, Y.; Lee, G.-H.; He, H.; Hsu, C.-Y.; and Katabi, D. 2018. RF-based fall monitoring using convolutional neural networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3): 1–24.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 6450–6459.

Vahdani, E.; and Tian, Y. 2022. Deep learning-based action detection in untrimmed videos: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, F.; Gao, Y.; Lan, B.; Ding, H.; Shi, J.; and Han, J. 2023. U-Shape networks are unified backbones for human action understanding from Wi-Fi signals. *IEEE Internet of Things Journal*, 11(6): 10020–10030.

Yang, J.; Chen, X.; Zou, H.; Lu, C. X.; Wang, D.; Sun, S.; and Xie, L. 2023. SenseFi: A library and benchmark on deep-learning-empowered WiFi human sensing. *Patterns*, 4(3).

Yang, J.; Zou, H.; Jiang, H.; and Xie, L. 2018. Device-free occupant activity sensing using WiFi-enabled IoT devices for smart homes. *IEEE Internet of Things Journal*, 5(5): 3991–4002.

Yang, L.; Zheng, Z.; Han, Y.; Cheng, H.; Song, S.; Huang, G.; and Li, F. 2024. DyFADet: Dynamic Feature Aggregation for Temporal Action Detection. In *European Conference on Computer Vision (ECCV)*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv preprint arXiv:2010.04159*.