STADe: Sensory Temporal Action Detection via Temporal-Spectral Representation Learning

Bing Li, Haotian Duan, Yun Liu, Le Zhang, Wei Cui, Joey Tianyi Zhou

Abstract—Temporal action detection (TAD) is a vital challenge in computer vision and the internet of things, aiming to detect and identify actions within temporal sequences. While TAD has primarily been associated with video data, its applications can also be extended to sensor data, opening up opportunities for various real-world applications. However, applying existing TAD models to sensory signals presents distinct challenges such as varying sampling rates, intricate pattern structures, and subtle, noise-prone patterns. In response to these challenges, we propose a Sensory Temporal Action Detection (STADe) model. STADe leverages Fourier kernels and adaptive frequency filtering to adaptively capture the nuanced interplay of temporal and frequency features underlying complex patterns. Moreover, STADe embraces adaptability by employing deep fusion at varying resolutions and scales, making it versatile enough to accommodate diverse data characteristics, such as the wide spectrum of sampling rates and action durations encountered in sensory signals. Unlike conventional models with unidirectional category-to-proposal dependencies, STADe adopts a cross-cascade predictor to introduce bidirectional and temporal dependencies within categories. To extensively evaluate STADe and promote future research in sensory TAD, we establish three diverse datasets using various sensors, featuring diverse sensor types, action categories, and sampling rates. Experiments across one public and our three new datasets demonstrate STADe's superior performance over state-of-the-art TAD models in sensory TAD tasks. *Code, models, and data will be released.*

Index Terms—Temporal Action Detection, Sensory Data Representations, Sensory Temporal Learning

1 Introduction

Temporal action detection (TAD) aims to accurately localize and identify actions within a temporal sequence, which is an essential and formidable task in many established fields such as computer vision and internet of things. The study of TAD originates from the computer vision community [1], where the increasing prevalence of video data such as surveillance raises crucial requirements to automatically analyze and understand temporal dynamics. While TAD is often associated with video, the concept has been extended to other temporal data, including sensor data [2]. In sensor data, actions or events can be represented as patterns or sequences of sensor readings over time, and the task is to localize and recognize specific actions or events of interest within the sensor data. TAD of sensory signals can be applied in many domains, such as human activity recognition [3], [4], environmental monitoring [5], and industrial automation [6].

Recently, deep neural models have emerged as the dominant approach in video TAD. Models like SSN [7] and AFSD [8] have been established as the *de-facto* standard. They utilize expressive deep neural backbones such as I3D [9] and X3D [10] to generate features and employ various decoding strategies (*e.g.*, actionness-grouping and anchor-based methods) to localize and identify potential actions. Their proficiency in capturing meaningful patterns has led to impressive performance in video TAD. However, the exploration of deep neural power in sensory TAD is still in its nascent stages. Existing sensory TAD models predominantly adhere to traditional methodologies such as policy-based noise removal, threshold selection, and Change Point Detection (CPD) methods [11] for activity segmentation. These *ad-hoc* methods are closely coupled with domain knowledge specific to the data or tasks at hand. Consequently, they lack enough flexibility to detect different action patterns, leading to inferior performances compared to their video TAD counterparts.

Despite the remarkable power and effectiveness of deep neural video TAD models, transitioning them from video TAD to sensory TAD presents non-trivial challenges. A fundamental distinction in understanding sensory TAD compared to videos lies in its intrinsic *temporal-spectral patterns*. Sensory signals, such as wireless or acoustic signals, fundamentally demonstrate wave-based properties [12]. This is in contrast to videos, where each frame captures a specific moment, and actions unfold gradually, revealing temporal-spatial arrangements. The wave nature of sensory signals not only introduces spectral details induced by the Doppler effect of physical movement but also contains *temporal-spectral patterns* in the form of frequency shifts (spectral variations) over time. These patterns illustrate the intricate interplay between temporal and spectral features. For instance, sudden or gradual transitions in the temporal domain influence the spectrum by introducing new frequencies or altering existing ones. Rapid temporal changes lead to a broad spectrum with energy spread across multiple

- *B. Li and L. Zhang are with School of Information and Communication Engineering, the University of Electronic Science and Technology of China.*
- *W. Cui is with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore.*
- *H. Duan is with Shandong University of Science and Technology*
- *Y. Liu is with the College of Computer Science, Nankai University, Tianjin 300350, China.*
- *J. T. Zhou are affiliated with both the Institute of High-Performance Computing (IHPC) and the Centre for Frontier AI Research (CFAR) at the Agency for Science, Technology and Research (A*STAR) in Singapore.*

TABLE 1
Primary distinctions between video TAD and sensory TAD.

Task	Characteristic			
	Feature	Frame rate	Propagation	SNR
Video TAD	Temporal-Spatial	Approx. 25	Instant	High
Sensory TAD	Temporal-Spectral	Varied	Delay	Low

frequency bands. Therefore, sensory TAD is supposed to have a notable emphasis on temporal-spectral patterns, where effectively capturing relevant information requires the adaptive fusion of temporal and spectral features.

The unique pattern schema, apart from its hardness, offers advantages for sensory TAD. In this schema, the start and end positions of actions are represented by changes in speed, and different action types exhibit varying speeds. While the success may rely on the interdependence of spectral and temporal aspects. On one hand, spectral data is analyzed relative to the temporal duration (*i.e.*, proposals) for computational purposes. Understanding the dynamics of action types in terms of their start and end dynamics is crucial for detecting proposals.

Adding on the temporal-spectral pattern schema, certain characteristics (summarized in Table 1) in aspects such as data sampling and signal propagation directly influence the temporal-spectral patterns of sensory data, further complicating an effective sensory TAD solution:

❶ *Low signal-to-noise ratio*. Sensory signals typically exhibit higher levels of noise compared to vision data, often characterized by a low signal-to-noise ratio (SNR) [13]. In contrast, video data frequently contains well-defined objects that can be distinguished from background noise. The prevalent noise and interference in sensory signals introduce an extra layer of complexity to the analysis, rendering the identification of meaningful patterns more challenging.

❷ *Diverse sampling rates*. Unlike the standard 25 fps frame rate typical in video data, sensory data usually spans a broad spectrum of sampling rates, ranging from under 1 Hz (*e.g.*, CO2 sensors) to over 1,000 Hz (*e.g.*, WiFi signals). This wide-ranging sampling frequency brings extra hardness in temporal granularity, spectral resolution, and time complexity, necessitating the model's adaptability. Specifically, a prime challenge is how to manage temporal scale due to differences in frame rates and action durations. Current TAD methods, such as snippet-level grouping strategies [14], [15], are predominantly tailored for video data with a standard intermediate frame rate. This inherently lacks the flexibility to handle transitions between high and low frame rates. Furthermore, the variations in scale pose challenges in selecting the appropriate spectral resolution, following Fourier uncertainty principles [16]. This principle implies that a signal cannot possess arbitrary precision simultaneously in both the time and frequency domains. Lastly, ultrahigh-frequency data (*e.g.*, exceeding 1,000 Hz like WiFi data) typically accompanies extended temporal lengths. Consequently, this results in prohibitively high time consumption for existing models, such as actionless grouping models like SSN [7] and BSN [17], due to their quadratic computational complexity of $\mathcal{O}(n^2)$.

❸ *Impacts of propagation*. Unlike optical signals, which are real-time, signals from other sensors (*e.g.*, acoustic devices) traveling through the medium may have delays, reflection, absorption, or scattering. This gives rise to potential temporal lags and blurred boundaries due to factors such as the echo effect [18]. As a result, rather than aligning precisely with a single time instance, a reading may also contain the "residuals" of historical moments (*i.e.*, historical dependency of boundaries). Compounded by the absence of a coherent definition concerning the temporal extent of an action [19], this aggravates the hardness of precisely determining the start and end points of an action, where the dynamics can become indistinct and overlapping, further complicating the analysis.

In this paper, we propose a *Sensory Temporal Action Detection* (STADe) model, which strategically combines the strengths of video TAD methodologies with unique insights from sensory data to facilitate sensory TAD tasks. STADe leverages the Aligned Temporal-Spectral Encoding (ATSE) backbone to enable effective representation learning, adaptively capturing the intricate interplay between temporal and spectral features. The ATSE backbone preserves the advantages of temporal-spatial representations of 3D backbones (*e.g.*, I3D), meanwhile, it seamlessly incorporates spectral information with Fourier kernels being compatible and alignable with the convolution-based temporal-spatial features. Considering the spectral domain resembles signal decomposition, we use an adaptive frequency filtering mechanism to eliminate background noise (trait ❶) in a trainable manner. To address the challenge of variances in scales introduced by diverse sampling rates (trait ❷), we capture and fuse temporal-spectral features at varied spectral resolutions and temporal scales. To facilitate prediction accuracy and rectify blurred boundaries resulting from the propagation (trait ❸), we introduce novel dependencies beyond the conventional *category* $\rightarrow$ *proposal*. These include bidirectional dependencies between proposals and categories, as well as historical dependencies within categories. Our pioneering approach diverges from conventional methods by introducing a cross-cascade predictor that facilitates TAD predictions through cross-cascade, enabling the simultaneous optimization of proposal generation and action recognition. Furthermore, we build three diverse datasets using various sensors, featuring diverse sensor types, action categories, and sampling rates, enabling the thorough training and evaluation of sensory TAD models. We will make these datasets publicly available to facilitate future research in this field.

The main contributions of this paper can be summarized as follows:

- This paper introduces the STADe model, specifically designed for sensory Temporal Action Detection (TAD) tasks. The model's temporal-spectral representation learning effectively combines temporal-spatial representations akin to the 3D backbones, seamlessly integrating spectral information in a compatible and alignable manner. This integration retains the benefits of both temporal-spatial and spectral representations.
- The paper proposes a novel cross-cascade predictor

that simultaneously enhances proposal generation and action recognition, deviating from traditional sequential or separated approaches. This innovation enhances the performance of the model in sensory TAD tasks.
- To address evaluation gaps in sensory TAD, the paper contributes three diverse datasets utilizing various sensors, including smartphone-embedded sensors and Wi-Fi Channel State Information (CSI). These datasets exhibit a wide array of sensor types, action categories, and sampling rates, enabling comprehensive assessments of sensory TAD methods.
- Extensive experiments demonstrate the superiorities of our model over state-of-the-art baselines on sensory TAD tasks.

2 Related Works

Traditional video TAD algorithms usually rely on tracking manually designed human motion features. For instance, DT [20] and iDT [21] track essential feature points (*e.g.*, static visual features such as HOG [22], and temporal-visual features like gray-level changes, contours, and the human body's skeleton) within a video's temporal region. Due to the limited expressiveness of hand-crafted features, these models often lag behind their deep-learning counterparts.

In contrast to traditional TAD methods' hand-crafted features, the features of deep-learning-based TAD models are automatically extracted via various feature representation backbones, such as CNN-based 2D-CNN [23], I3D [9], and transformer-based DETR [24]. TAD can be naturally decomposed into two sub-tasks, *i.e.*, action localization and recognition. According to the ways of dealing with the two sub-tasks, existing deep learning models can be further classified as sequential models and one-stage models, which are reviewed in §2.1 and §2.2, respectively.

2.1 Sequential Models

Sequential models follow a "localization-then-recognition" paradigm, which generates actions' beginning and end timestamps (*a.k.a.* proposals) before making recognitions. Sequential models typically emphasize proposal generation more as the classification hinges greatly on accurate proposals. According to the strategy for generating proposals, the methodologies can be further classified into actionness grouping methods and anchor-based methods.

Actionness grouping methods create a complete proposal by aggregating frame- or snippet-level proposal segments [25]. This involves post-processing the actionness information of fine-grained snippets to construct action proposals. S-CNN [26] employs fixed-sized sliding windows to detect potential proposal segments and applies non-maximized suppression (NMS) to eliminate overlapping segments. TAG [27] decides on the snippet level for each snippet and groups adjacent snippets to form a complete proposal. SSN [7] divides coarse proposals into three semantic segments and independently learns them, predicting probabilities of activity and completeness. BSN [17] initially identifies temporal action segment boundaries, forming proposals by collecting intervals with high start and end probabilities and filtering out low-confidence intervals. This framework is later enhanced into BMN [28], which generates a Boundary-Matching confidence map to improve proposal quality. SEP [11] is a proposal detection method specialized for sensory signals. SEP proposes an unsupervised change point detection algorithm that identifies key time points exhibiting significant distribution shifts.

Another type of sequential model is anchor-based models, such as TURN [29], R-C3D [14], and GTAN [30], which treat the proposal generation as a temporal regression problem by adjusting pre-defined anchors. TURN [29] aggregates features from basic video units to create clip-level features, which are then used for activity classification and temporal boundary regression. Similarly, R-C3D [14] draws inspiration from the Faster R-CNN [31] approach, involving proposal generation, proposal-wise pooling, and final prediction. GTAN [30] modifies the pooling procedure by incorporating a weighted average using a learnable Gaussian kernel to adjust the temporal scale of every action proposal. These methods, however, have a limitation due to their fixed pre-defined anchors, making them less flexible when dealing with different action classes. In contrast, our model eliminates the need for fine-tuning additional hyperparameters for anchors, resulting in greater efficiency.

2.2 One-stage Models

The sequential framework is easy to suffer the error propagation problem as proposal generation and classification have to be trained separately. The one-stage paradigm is proposed to train action localization and recognition in a joint manner to solve the two sub-tasks simultaneously. SSAD [15] directly makes frame/snippet-level classification and groups neighboring frames under the same action category. However, when dealing with long-range dependencies, snippet-level predictions may not adequately capture frame-wise relationships. To address this, Coarse-Fine [32] employs a two-stream architecture with distinct temporal resolutions (coarse and fine streams) to capture long-term motion information. Frame-level category grouping methods indirectly predict action boundaries, which may not be suitable for actions of varying durations. Recent research has introduced an anchor-free detection approach to tackle this challenge. Instead of predicting only action categories, anchor-free techniques predict both categories and boundaries for each snippet. SS-TAD [33] outputs a triplet specifying the action's start, end, and category for a snippet. To reduce redundant predictions, SS-TAD employs non-maximum suppression (NMS) to prune excessive proposals. Extending the SS-TAD framework, AFSD [8] improves it by incorporating a saliency-based proposal boundary feature and introducing a refined stage to correct any discrepancies in the initial predictions. TALLFormer [34] utilizes a short-term transformer for encoding short-term actions and includes a long memory module to capture longer-duration actions.

However, the majority of existing TAD models are primarily tailored for video data. When applied to sensory TAD, these models encounter inherent limitations owing to the distinctive challenges posed by sensory signals. These challenges include the intricate temporal-spectral patterns

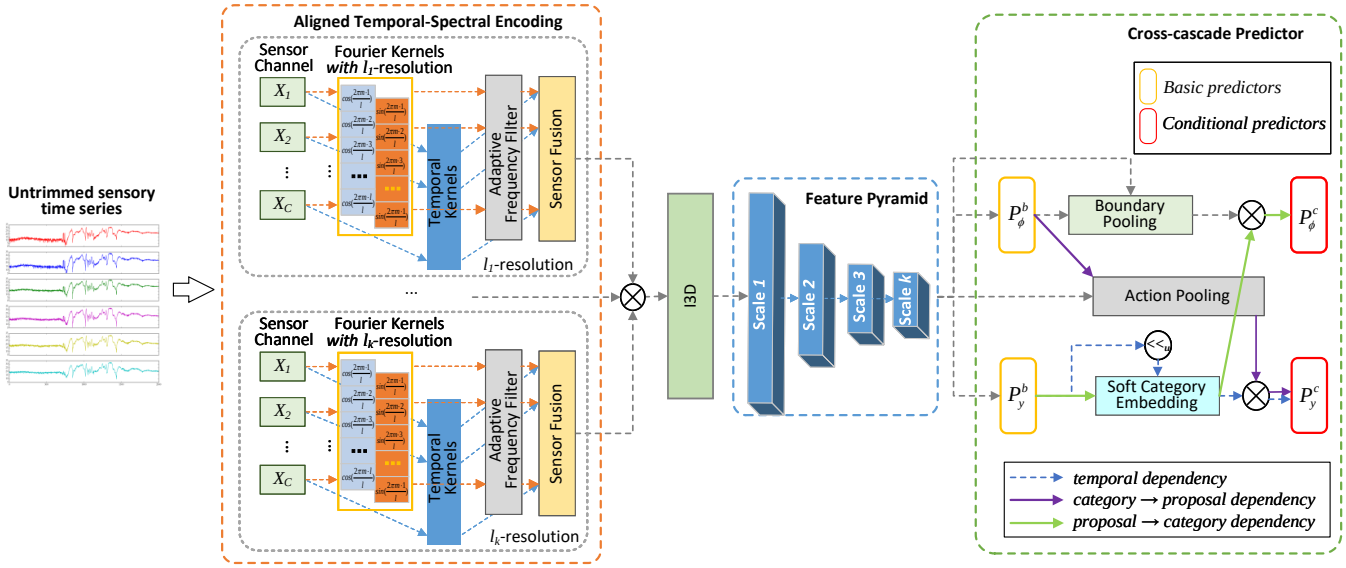


Fig. 1. Architecture of the proposed STADe model. Initially, an untrimmed sensory time series is inputted into an *Aligned Temporal-Spectral Encoding* module represented by a dotted-orange box. This module is designed to extract temporal-spectral features and employs multi-resolutions (l_1 to l_k) with varying kernel sizes to overcome limitations in spectral resolution. Then, the features from these resolutions are concatenated and then passed through the Inflated 3D ConvNet (*I3D*) to extract spatial information. Additionally, a *Feature Pyramid* incorporating multiple scales is utilized to adapt to varying frame rates and granularities. Lastly, an anchor-free *Cross-cascade Predictor*, comprising both basic and conditional predictors, facilitates accurate proposals and category predictions.

and are further compounded by the diverse sampling rates, effects of propagation, and persistent noise. Thus, there is a clear need for dedicated methods that exceed the capabilities of traditional video TAD approaches to effectively address these challenges in sensory TAD.

Another line of work related to TAD is action recognition, which involves determining the action category of a trimmed video or sensor snippet that typically contains a single action. For example, THAT [13] proposes a Transformer-based action recognition model leveraging WiFi signals. Considering the labor-intensive nature of video labeling, Liu et al. [35] introduced Deep Image-to-Video Adaptation and Fusion Networks (DIVAFN), which enhance action recognition in videos by transferring knowledge from images, using video keyframes as a bridge. SAKDN [36] further improves action recognition in the video modality by adaptively transferring and distilling knowledge from multiple wearable sensors. To enhance representation learning for action recognition, other works [37], [38] propose approaches such as temporal contrastive graph learning [37] and self-supervised learning on skeleton sequences [38]. However, action recognition differs from TAD in that TAD requires identifying both the start and end positions, as well as the action category, within an untrimmed sequence. This makes TAD a more challenging task, often requiring additional efforts for improved accuracy.

2.3 Sensory Temporal Action Detection vs. Visual Temporal Action Detection

Compared with visual TAD, sensory TAD offers several unique advantages:

i) Context-specific nature. STADe addresses sensor-specific challenges that are absent in video data. Unlike video-based TAD, which captures spatial information through well-defined frames, sensory data is often unstructured and consists of continuous signal streams (e.g., from wireless or embedded sensors) with inherent temporal-spectral dependencies. This distinct nature requires STADe to extract meaningful patterns from signals that lack clear object definitions and aligned frames.

ii) Flexibility across sensor types. The task of sensory TAD often involves various sensor types (e.g., Wi-Fi, accelerometers, gyroscopes). While video-based methods rely on clear, recognizable objects from camera, sensor data varies in signal type, dynamic range, and sampling frequency. STADe's adaptive design allows it to process these variations, offering enhanced flexibility and generalizability across different modalities.

iii) Real-world applicability in privacy-sensitive domains. Sensory TAD is well-suited for applications such as human activity recognition, healthcare, environmental monitoring, especially where visual data is unavailable or impractical. In privacy-sensitive scenarios, cameras may capture user bio-information, raising significant privacy concerns. In contrast, sensor-based monitoring (using Wi-Fi or motion sensors) preserves privacy while still enabling effective action detection.

iv) Adaptability to low-quality signals. Sensory signals tend to be noisier and less structured than visual data. STADe employs mechanisms like adaptive frequency filtering and bidirectional dependencies to address low signal-to-noise ratios (SNR) and other noise-related challenges, ensuring robust action detection even in suboptimal conditions.

3 Methodology

3.1 Problem Definition

Sensory TAD identifies the proposal (*i.e.*, start and end timestamps), as well as the category of each action in-

stance in an untrimmed sensory time series. The dataset $\mathcal{D}$ for sensory TAD contains n records $\{d_1, d_2, ..., d_n\}$, where each record $d_i = \{X, \Psi\}$ consists of a sensory time series $X \in \mathbb{R}^{C\times T}$ with T time points and C sensor channels, along with the associated action annotations Ψ. The annotations include M_x tuples $\{(\phi_m, y_m)\}_{m=1}^{M_x}$, where each tuple denotes the ground-truth annotation of an action instance. Specifically, $\phi_m = (\psi_m, \xi_m)$ represents the start and end position of the action proposal, and y_m indicates the action category. M_x represents the total number of action instances in the time series X. Our goal is to train a model that predicts action proposals and their respective action categories with high recall and precision, consistent with the ground truth in the test set.

3.2 Model Overview

The framework of our model is shown in Fig. 1. It takes an untrimmed sensory time series as input and uses an aligned temporal-spectral encoding module to extract informative features in both temporal and spectral domains. To address limitations related to spectral resolution, we use multi-resolutions ranging from l_1 to l_k, incorporating kernels of varying sizes. By concatenating the features from all k resolutions, we pass them through the Inflated 3D ConvNet (I3D) [9] to extract deep semantic information, leveraging its high expressiveness. To accommodate varying frame rates and temporal granularities, we utilize a feature pyramid that consolidates features at different temporal scales. Finally, the anchor-free cross-cascade predictor, comprising basic and conditional predictors, enables predictions of proposals and categories.

3.3 Aligned Temporal-Spectral Encoding

As the Doppler effect, sensory time-series signals often contain valuable frequency-related characteristics that convey the variations in the observed phenomena. Inspired by Fourier initialized convolution [39], we propose an Aligned Temporal-Spectral Encoding (ATSE) to effectively encode both temporal and spectral features and their intricate interplay within sensory data, facilitating the extraction of informative features. Within the ATSE module, our proposed learnable Fourier kernels (§3.3.1) and the adaptive frequency filter (§3.3.2) are specially designed to eliminate background noise for tackling the trait ❶ in §1; the temporal kernels (§3.3.3), sensor fusion (§3.3.4), and multi-resolution composite (§3.3.5) are anticipated to learn multi-scale features from different perspectives (*i.e.*, temporal, sensory, and multi-resolution perspectives) for addressing the trait ❷.

3.3.1 Fourier Kernels

Conventional Fourier transform lacks timestamps and cannot capture the interplay with temporal features. Short-Time Fourier Transform (STFT) is not trainable and lacks the flexibility to adapt to diverse spectrum distributions. Hence, we innovate by introducing Fourier kernels, which are aligned with convolution-based temporal features, and are trainable to adapt to varying spectrum patterns.

For a single sensor channel $x \in \{X_1, X_2, ..X_C\}$, we convolve it with kernels $\boldsymbol{w}^S \in \mathbb{R}^{k\times l}$ $(k \le l)$[1], where the i-th value of the m-th kernel is defined as

$$\boldsymbol{w}_{[m,i]} = e^{-j\frac{2\pi mi}{l}}, \tag{1}$$

where l is the kernel size and j denotes the imaginary unit. Notably, as convolving a signal with a sliding window, the kernels of Eq. 1 establish equivalence to STFT, *i.e.*, $\text{STFT}(x)_{[m,n]} = \sum_{i=1}^{l} x_{[n+i]} e^{-j\frac{2\pi mi}{l}}$, in that the m-th kernel extracts the component at frequency $\frac{m}{l}$ within a l-length time slice starting from position n.

Eq. 1 inevitably introduces complex values, adding implementation difficulties to neural networks. To represent the Fourier transform in terms of real-valued convolutions, we separate the real and imaginary parts as two separate kernels, $\boldsymbol{w}^r$ and $\boldsymbol{w}^i$, where the i-th value of the m-th kernel is given by

$$\boldsymbol{w}^r_{[m,i]} = \cos\left(\frac{2\pi mi}{l}\right), \boldsymbol{w}^i_{[m,i]} = -\sin\left(\frac{2\pi mi}{l}\right), \tag{2}$$

where the range of m falls in $[0, l/2]$ as being conjugate symmetric for the Fourier transform of a real-valued series. Based on the Fourier kernels of Eq. 2, we can compute the amplitude $|x| \in \mathbb{R}^{k\times T}$ and phase $\angle x \in [-\pi, \pi]^{k\times T}$ to enhance it with physical meaning:

$$|x| = \sqrt{(x \circ \boldsymbol{w}^r)^2 + (x \circ \boldsymbol{w}^i)^2}, \angle x = tan^{-1}(\frac{x \circ \boldsymbol{w}^i}{x \circ \boldsymbol{w}^r}), \tag{3}$$

where $\circ$ is the convolution operation.

3.3.2 Adaptive Frequency Filter

Although spectral analysis using Eq. 2 can reveal important features in the frequency domain, noise components, such as high-frequency noise, are also preserved during the Fourier transform as revealed by Parseval's theorem. To address this issue, we propose a frequency gate that serves as an adaptive filtering mechanism to reduce noise and eliminate meaningless frequency components, thereby prioritizing useful frequency components over others. The final encoding of the spectral stream H^S (having $2k \times T$ size) is obtained by concatenating the amplitude and phase parts as follows:

$$H^S = [\boldsymbol{\beta}|x|; \boldsymbol{\beta}\angle x], \boldsymbol{\beta} = \sigma(B), \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $B \in \mathbb{R}^{1\times k}$ is a learnable parameter vector.

3.3.3 Temporal Kernels

Temporal features are extracted using standard convolution kernels, $\boldsymbol{w}^T \in \mathbb{R}^{k\times l}$, resulting in feature maps $H^T = x \circ \boldsymbol{w}^T$. The temporal-spectral encoding is obtained by aligning the time dimension of temporal featureH^T and spectral feature H^S, yielding the final encoding $H^F = [H^T; H^S]$, $H^F \in \mathbb{R}^{3k\times T}$. The kernels in both the spectral and temporal streams share the same shape (in terms of number and size)[2], enabling them to capture both spectral and temporal

1. The proposed kernels can be trained, as in FIC [39], to facilitate flexibility and generalizability across different tasks, or kept fixed during training to enforce strict Fourier patterns.

2. The feature maps are $\lfloor (l-1)/2 \rfloor$ zero-padded on both sides to ensure identical temporal size.

Fig. 2. Dependency between category and proposal. y_{his}: historical categories prior to the position i; y_i: the i-th category; ϕ_i: the i-th proposal.

features that align with the same time step. This allows for multi-scale deep fusion without losing temporal order, which is crucial for TAD tasks.

3.3.4 Sensor Fusion

The stacking of all C channels' aligned temporal-spectral encodings can be represented as $H \in \mathbb{R}^{3kC\times T}$. To reduce redundancy among sensor channels, we fuse and combine the features across them by utilizing kernels $\boldsymbol{w}^{sf} \in \mathbb{R}^{C'\times 3kC}(C' \leq C)$, to convolve on H along the temporal axis $F^{sf} = \boldsymbol{w}^{sf} \circ H \in \mathbb{R}^{C'\times T}$.

3.3.5 Multi-Resolution Composite

The Fourier uncertainty principles [16] dictate that a signal cannot have arbitrary precision in both temporal and spectral domains simultaneously. To ensure judicious and effective utilization of potential resolutions and to avoid the limitations of using a single window size, we capture signals at different resolutions by employing kernels of various sizes l. This enables the network to bring the precision of temporal and spectral features together by automatically determining suitable windows and their feature combinations during the training.

3.4 Deep Fusion at Multi-scale

3.4.1 I3D backbone

The outputs obtained from each resolution are treated as separate channels and concatenated to create a composite feature map, which is further processed by the I3D backbone [9], a prevailing backbone network that is commonly used for temporal analysis tasks, *e.g.*, video classification, speaker recognition, and cross-modality retrieval, to extract informative features.

Owing to its sparse network architecture, I3D exhibits strong expressive power while maintaining a low risk of overfitting and computational costs. This make I3D a *de-facto* standard for learning temporal-spatial representations in many state-of-the-art models, such as AFSD [8], which has demonstrated cutting-edge performance in learning temporal and spatial patterns from sensor data. Furthermore, empirically, we found that 3D networks like I3D significantly outperform traditional 2D networks, such as basic CNN and ResNet. When compared with other 3D convolutional networks like C3D [40], R3D [41], and R(2+1)D [42], I3D exhibits very similar performance while being more efficient.

3.4.2 Multi-Scale Feature Pyramid

In response to varying frame rates and temporal granularity of sensor data (trait ❷ in §1), we further attempt to capture effective patterns across multiple temporal scales by utilizing a feature pyramid structure, as depicted in Fig. 1. Each layer in the pyramid corresponds to a specific scale, which is controlled by utilizing 1D convolutions with different kernel sizes k and stride s. From the bottom to the top, the temporal length of the output features $F^i \in \mathbb{R}^{T^i\times C'}$ at the i-th layer is half of that of the layer $i-1$, *i.e.*, $T^i = T^{i-1}/2$, resulting in a larger receptive field and temporal scale.

3.5 Anchor-free Cross-cascade Predictor

Apart from predicting proposals, a classifier is employed to anticipate the action category. Typically, the classification component is applied either after generating proposals or learned independently. The latter is then augmented by a correction step to integrate it with the proposal information in recent research [8]. Unfortunately, previous models only account for a one-way *category* $\rightarrow$ *proposal* relationship. We propose the presence of two additional dependencies within sensory TAD (illustrated in Fig. 2) that could significantly enhance the task but have not been explored in existing methods, thus settling the trait ❸ in §1. *i*) The first is the bidirectional dependency between category and proposal, suggesting an extra *proposal* $\rightarrow$ *category* connection indicating that the proposal is influenced by its category. For example, knowing that an action belongs to a specific category (such as walking or running) helps to better define the start and end points of the action proposal. In the case of "walking" and "running", the duration and speed of the action provide vital clues for accurate proposal localization. From a *spectral perspective*, the temporal patterns associated with different actions exhibit distinctive variations (e.g., speed changes or acceleration), and knowing the category helps to refine these temporal boundaries[3]. *ii*) The second is the *temporal dependency* among categories, implying that categories frequently follow a specific order. The *temporal dependency* allows the model to better understand how actions relate to each other over time. For instance, "sitting" often precedes "standing", and the detection of one can help inform the detection of the other. This temporal relationship helps mitigate errors introduced by overlapping actions or the residuals of past actions[4].

It is worth noting that, the two mentioned dependencies establish a link between proposal ϕ_i and the past categories $y_{i-1}, ..., y_1$ through the Markov chain. This is especially vital for sensory signals, considering that many sensory signals, such as wireless or acoustic signals, often experience time delays or signal overlap due to propagation in a medium or echoes caused by obstacles. In this context, we propose to predict the joint distribution $p(\phi, y|F)$ of the proposal ϕ and category y. From a probabilistic perspective, given the bi-directional dependency, this comprises a cyclic Bayesian network (CBN). Generally, solving the CBN is challenging due to the presence of feedback loops. Some solutions to this challenge include the junction tree algorithm [43] or Markov chain Monte Carlo [44], albeit these approaches are computationally intensive.

Given that our CBN is not excessively complex, we suggest employing a straightforward path sampling method to approximate it. This approach can be viewed as a simplified

3. Ablating the *proposal* $\rightarrow$ *category* dependency results in a 0.02 point decrease in mAP across all datasets (Table 4).

4. Ablating *temporal dependency* leads to an average decrease of 0.018 in mAP across all datasets (Table 4).

variant of the junction tree algorithm [43]. Specifically, we sample two paths from the CBN, outlined as follows:

$$\begin{aligned} p_1 &: p(\phi, y|F) = p(\phi|F, y)p(y|F), \\ p_2 &: p(\phi, y|F) = p(y|F, \phi, \tilde{y})p(\phi|F)p(\tilde{y}|F), \end{aligned} \tag{5}$$

Where $\tilde{y}$ represents the sequence of categories from past timestamps. The p_1 path[5] follows the conventional two-step generative process to generate the proposal ϕ first, and the p_2 path explicitly explores the *proposal* $\rightarrow$ *category* and temporal dependencies.

By integrating p_1 and p_2, the objective is to get

$$\begin{aligned} \phi^\star, y^\star &= \arg\max_{\phi,y} p(\phi, y|F) \\ &= \arg\max_{\phi,y} [\log p(\phi|F, y) + \log p(y|F) + \\ &\quad \log p(y|F, \phi, \tilde{y}) + \log p(\phi|F) + \log p(\tilde{y}|F)]. \end{aligned} \tag{6}$$

Given the computational complexity associated with the continuity of ϕ, we adopt an approximation strategy as follows:

$$\begin{aligned} \phi^\star &\approx \arg\max_{\phi} \left[\log p(\phi|F, \hat{y}) + \log p(\phi|F)\right], \\ y^\star &\approx \arg\max_{y} \left[\log p(y|F) + \log p(y|F, \hat{\phi}, \tilde{y})\right], \\ \hat{\phi} &= \arg\max_{\phi} p(\phi|F), \quad \hat{y} = \arg\max_{y} p(y|F). \end{aligned} \tag{7}$$

This forms a hard expectation-maximization (hard-EM) algorithm. Under certain conditions[6], the algorithm is guaranteed to converge to a local optimum by coordinate ascent [45].

3.5.1 Design Details of the Predictor

Eq. 7 in the sensory TAD context is formulated to implement two basic predictors $P_y^b = p(y|F)$ and $P_\phi^b = \mathbb{E}(\phi|F)$, as well as two conditional predictors $P_y^c = p(y|F, \phi, \tilde{y})$ and $P_\phi^c = \mathbb{E}(\phi|F, y)$ on both pyramid features and outputs of basic predictors. Before discussing the details of neural models, it is essential to consider the strategies used for predicting proposals. The choice of strategy can significantly impact the model's effectiveness, particularly when dealing with diverse sensor frame rates and high-frequency signals.

Anchor-free Proposal Prediction. Predicting proposals is typically in either an action-based or anchor-based manner. We argue that the two strategies are not well-suited due to the highly diversified ranges in frame rates of sensors. The exhaustive search (with $\mathcal{O}(T^2)$ complexity) in action-based methods (*e.g.*, SSN [7] and BSN [17]) is prohibitively expensive for high-frequency signals such as WiFi[7], which create highly dense and semantically blurred boundaries. Conversely, pre-defined anchor boxes in anchor-based methods [14] obviously lack flexibility for switching between high and low frame rates and are unsuitable for sensory TAD, which lacks a clear object/boundary or region of interest. We tackle the issue of varied frame rates and temporal durations in sensory TAD predictions by employing the anchor-free paradigm [8], [48]. Unlike anchor-based approaches, anchor-free TAD does not rely on pre-defined anchor boxes but instead predicts the relative distances $(\hat{d}_i^s, \hat{d}_i^e)$ between the start and end boundaries from the position i through regression.

To enhance the performance across a range of action durations, the relative distance is predicted independently for each scale or layer within the feature pyramid. On top of the v-th scale, the absolute start and end positions of proposal $\hat{\phi}_i$ can be inferred as $\hat{\psi}_i = i \times 2^v - \hat{d}_i^s$ and $\hat{\xi}_i = i \times 2^v + \hat{d}_i^e$, respectively.

Basic Predictors. The two basic predictors P_y^b and P_ϕ^b employ one temporal convolution layer that takes as input pyramid features F^i and is followed by convolution-based regression or classification heads. These heads are shared across all pyramid layers. Taking the proposal as an example, we have $\hat{\phi}_i = \text{Linear}(\text{Conv}(F))$.

Conditional Predictors. In P_ϕ^c, the category features are incorporated as a mixture of category embeddings, *i.e.*, *Soft Category Embedding*, guided by the output distribution of P_y^b:

$$\hat{F}^y = P_y^b \boldsymbol{w}^c, \tag{8}$$

where $P_y^b \in \mathbb{R}^{T \times n_y}$ represents the categorical distribution for all T temporal points, n_y is the number of categories, and $\boldsymbol{w}^c \in \mathbb{R}^{n_y \times d_y}$ is a set of trainable category embeddings with the dimension d_y.

As some work reveals that the features are more concentrated around span boundaries [8], [49], we adopt the *Boundary Pooling* [8] to generate salient features of the start and end boundary (denoted as $\hat{F}^s$ and $\hat{F}^e$, respectively) by leveraging the proposal prediction $\hat{\phi} = \{\hat{\psi}, \hat{\xi}\}$ of the basic predictor:

$$\hat{F}_{k,i}^s = \max_{j \in [\hat{\psi} - \frac{l_{\hat{\phi}}}{\sigma_a}, \hat{\psi} + \frac{l_{\hat{\phi}}}{\sigma_b}]} F_{k,i}, \qquad \hat{F}_{i,k}^e = \max_{j \in [\hat{\xi} - \frac{l_{\hat{\phi}}}{\sigma_a}, \hat{\xi} + \frac{l_{\hat{\phi}}}{\sigma_b}]} F_{j,i}, \tag{9}$$

where $l_\phi = \hat{\xi} - \hat{\psi}$ is the length of the proposal, and σ_a, σ_b are hyper-parameters used to adjust the ratio of regions selected both the outside and inside of the proposal, respectively.

The prediction is made by a simple CNN on top of concatenating boundary features and category features:

$$P_\phi^c = \text{CNN}([\hat{F}^s; \hat{F}^e; \hat{F}^y]). \tag{10}$$

To further improve performance through boosting, the predictor P_ϕ^c predicts the residual $\hat{\Delta}_\phi$ of $\hat{\phi}$. This enables it to work synergistically with the basic predictor.

For P_y^c, we extract action features within the proposal ϕ using *Action Pooling*, which contains temporal convolution and pooling like

$$\hat{F}^p = \text{Pooling}(\text{Conv}(F_{[\hat{\xi}:\hat{\psi}]})). \tag{11}$$

This underscores the significance of the proposal content in categorization, as it manifests temporally ordered, continuous movement. This stands in contrast to P_ϕ^c, which relies solely on boundary features. To incorporate historical categories $\tilde{y}$, we first get the category features $\hat{F}^y$ using Eq. 8. These features are then left-shifted by u steps along the temporal axis to produce $\hat{F}^{\tilde{y}} <<=_u \hat{F}^y$, employing left-truncation and zero-filling. This aligns the historical category features $\hat{F}_i^{\tilde{y}}$ with the position i, representing a

5. In $p1$, we disregard $p(y|F, \tilde{y})p(\tilde{y}|F)$ to sidestep the large computational cost of sequential decoding.

6. The loss function should be continuously differentiable with a Lipschitz continuous gradient.

7. A typical sampling/frame rate of WiFi signal is 1,000 Hz, in contrast to most video datasets (*e.g.*, Kinetics [46] and UCF101 [47]) with a frame rate of about 25 fps.

category from u steps prior, defaulting to $u = 1$. The proposal context features and historical category features are then concatenated and fed into the classifier to obtain the labels:

$$P_\phi^c = \text{CNN}([\hat{F}^p; \hat{F}^{\tilde{y}}]). \tag{12}$$

3.6 Training and Inference

In this part, we elaborate on the training and inference processes of the proposed STADe model.

Training samples. During training, a position i is marked as positive if i resides within a ground-truth proposal ϕ_j such that $\psi_j \leq i \leq \xi_j$. To prevent arbitrariness, the conditional predictor P_ϕ^c only processes positive examples with an Intersection over Union (IoU) between the ground-truth ϕ_j and $\hat{\phi}_j$ (predicted by P_ϕ^b) over 0.5.

Loss function. We adopt the following loss function:

$$\mathcal{L} = \alpha\mathcal{L}_\phi^b + \mathcal{L}_y^b + \alpha\mathcal{L}_\phi^c + \mathcal{L}_y^c, \tag{13}$$

where α is a loss balance factor. $\mathcal{L}_\phi^b = \frac{1}{N}\sum_i \mathbb{I}(y_i \geq 1)(1 - \frac{|\hat{\phi}_i \cap \phi_i|}{|\hat{\phi}_i \cup \phi_i|})$ is IoU loss [8] and $\mathcal{L}_\phi^c = \frac{1}{N_c}\sum_i \mathbb{I}(y_i \geq 1)(|\hat{\Delta}_i - \Delta_i|)$ is the L1 loss between the predicted residual and the true offset. $\mathcal{L}_y^b$ and $\mathcal{L}_y^c$ are both focal loss [50] of category.

Inference. For the i-th position in j-th layer of the feature pyramid, we get $\hat{\phi}_{i,j}$, $P_y^b(i,j)$, $\hat{\Delta}_{i,j}$, and $P_y^c(i,j)$ predicted by P_ϕ^b, P_y^b, P_ϕ^c, and P_y^c, respectively. Incorporating Eq. 7, the final predition is:

$$\phi_{i,j}^\star = \hat{\phi}_{i,j} + \hat{\Delta}_{i,j}, \quad y_{i,j}^\star = \arg\max_y \left[P_y^b(i,j) + P_y^c(i,j)\right].$$

To remove redundant proposals caused by the frame-level predictions, non-maximum suppression (NMS) [51] is utilized to select the most confident prediction from highly overlapped proposal predictions.

4 Datasets

In order to evaluate the effectiveness of sensory TAD models, we perform comprehensive experiments on four datasets, including the widely-used public dataset *DeepSeg* [2] and three self-collected datasets with diverse sampling rates, sensor types, and action types. The self-collected datasets aim to address the paucity of sensory TAD evaluation datasets and facilitate a comprehensive evaluation of sensory TAD methods. *SeBehave*, collected using our smartphone app, is a locomotion recognition dataset based on readings from smartphone-embedded sensors, including a three-axis accelerometer, a gravity accelerometer, and a gyroscope. A detailed description of *SeBehave* can be found in §4.1. *WiKeystroke* (detailed description in §4.2). and *WiBehave* (detailed description in §4.3) are both based on Wi-Fi Channel State Information (CSI) and differ in action types and sampling rates, with *WiKeystroke* recording keystrokes on a standard QWERTY keyboard (number keys) at 1,000 Hz and *WiBehave* recording locomotion with a sampling rate of 500 Hz. Details on the datasets are summarized in Table 2. We follow common settings in sensory segmentation [2], to take a 7:3 training and testing split for all datasets. We believe that our new datasets will facilitate future research in sensory TAD by enabling thorough training and evaluation of deep-learning-based models.

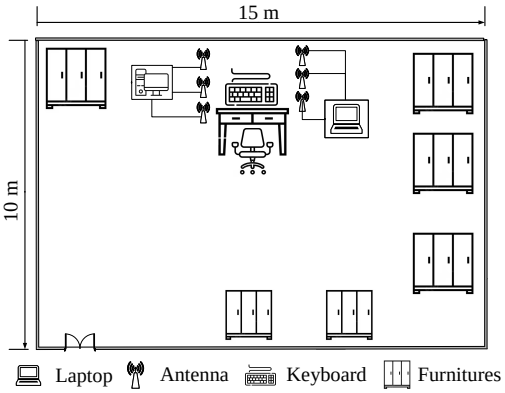


Fig. 3. Experimental environment layout of the *WiKeystroke* dataset.

4.1 SeBehave Dataset

SeBehave is a self-collected dataset for localizing and recognizing human locomotions using embedded sensors of the smartphone. We develop a smartphone APP based on Android OS to collect data in a daily scenario. The APP records data generated by the smartphone's tri-axial accelerometers (linear acceleration) and gyroscope (angular velocity) sensors, as well as the gravity accelerometer sensor along the (X, Y, Z) axes during various physical activities. The sampling frequency is set to 200 Hz.

The participants are four master student volunteers. Each volunteer is tasked to perform 4-6 randomly selected activities from a predefined set of seven activity types, which include *walking*, *running*, *standing*, *ascending stairs*, *descending stairs*, *lying down*, and *sitting*. To mark the start and end time of each activity, volunteers are required to tap the corresponding button on the mobile app to record the corresponding timestamps. Each performance is timed to last approximately one minute. During the execution of these activities, sensor data are continuously recorded using the designated mobile APP. Each volunteer completes this performance 250 times, resulting in 250 individual records per volunteer. In total, there are 1,000 records for all six volunteers. Notably, participants are free to hold the smartphone in either hand, and there are no constraints on the duration of each activity. Nonetheless, attempts are made to maintain a balanced distribution of the seven different activities within the dataset.

Each record has the dimension of $[T \times 200, 9]$, where T represents the time (in seconds) for this record, "200" is the sampling rate, and "9" denotes the nine sensor channels, including tri-axial accelerometers, angular velocity, and gravity acceleration.

4.2 WiKeystroke Dataset

WiKeystroke is a self-collected dataset for localizing and recognizing keystroke actions using WiFi signals. We develop a prototype system to replicate typical keyboard strokes of typing. The layout of the data collecting scenario is shown in Fig. 3. The room is 15×10 meters in size, with furniture such as closets. A desk is placed in the room for placing a mouse, a keyboard, a camera, and speakers (used as a timing marker for subsequent dataset labeling). The equipment consists of a laptop equipped with an Intel 5300 NIC, a

TABLE 2
Statistics of the datasets used in our experiments.

	DeepSeg	*WiBehave*	*WiKeystroke*	*SeBehave*
Sampling Rate	50Hz	500Hz	1,000Hz	200Hz
# Time Series	150	500	174	1,000
Avg. Length	8,000	8,500	8,000	12,000
# Categories	10	7	10	7
# Avg. Act.	10	4	4	5
Sensor Type	WiFi CSI	WiFi CSI	WiFi CSI	Smartphone sensors
Actions	hand swing, hand raising, pushing, drawing O, drawing X, boxing, picking up, running, squatting, and walking	walking, running, jumping, waving, bending, sitting, and standing up	37 keys (digits 0-9, letters a-z, and the space key) on a QWERTY keyboard	walking, running, standing, ascending stairs, descending stairs, lying down, and sitting

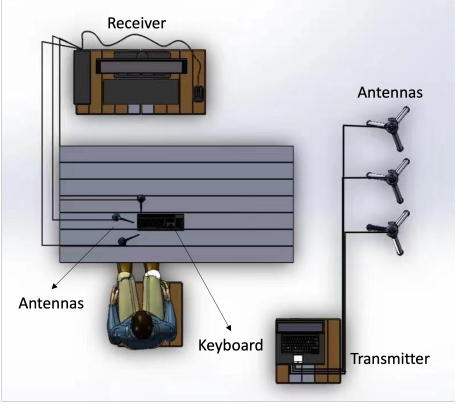


Fig. 4. Detailed placement of the keyboard and WiFi transmitter and receiver in the *WiKeystroke* dataset.

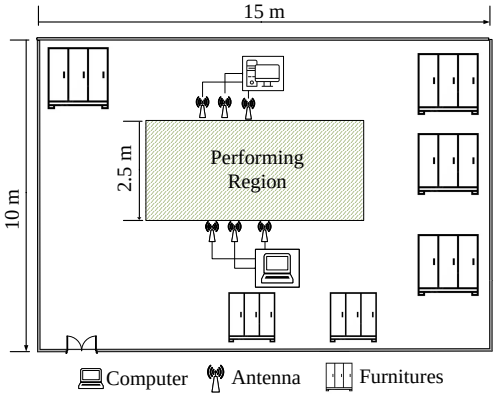


Fig. 5. Experimental environment layout of the *WiBehave* dataset.

desktop computer, and antennas. The laptop serves as the transmitter, while the desktop computer acts as the receiver. The transmitter sends WiFi signals through antennas, and the receiver collects CSI data using the CSI-Tool[8] software installed on the Ubuntu 14.04 system. The antennas of the receiver are placed on the desk, and the transmitter is placed at a height of approximately 1.5m using tripods. To mitigate potential interface disruptions affected by inferior Tx and Rx placements on subtle keystroke movements, we meticulously optimize Rx-Tx positions for improved keystroke detection. This optimization aligns with the Fresnel zone model [52] of WiFi signal propagation, validated through practical experiments. To account for potential detection limitations with a single Rx-Tx antenna pair, we utilize three pairs of antennas dispersed in different locations. This diversified setup allows the data collection platform to observe keystroke actions from various angles by forming interleaved Fresnel zones. This enhances WiFi signal sensitivity and precision in perceiving keystroke actions, enabling a consistent mapping between keystroke actions and CSI data patterns for effective keystroke recognition. Through practical experiments, we set the layout of the platform as depicted in Fig. 3. The antennas closest to the keyboard are receiving antennas, numbered from bottom to top as (0), (1), and (2). Antenna (0) is positioned 20cm from the keyboard, antenna (1) is 20cm from the keyboard, and antenna (2) is 7cm from the keyboard. The receiving antennas have a bend angle of 120 degrees. The remaining three antennas are transmitting antennas, numbered from bottom to top as (0), (1), and (2), with a 40cm spacing and the closest distance to the keyboard being 120cm. The schematic diagram of the keystroke collection platform is illustrated in Fig. 4. Both the transmitter and the receiver have three commercial 5dB omnidirectional antennas. Each antenna has 30 subcarriers and operates at 2.4 GHz with a 20 MHz channel bandwidth, with a sampling frequency of 1,000 Hz.

One student volunteer participates in the study and is instructed to press the keys corresponding to press 37 keys (digits 0-9, letters a-z, and the space key) on a standard QWERTY keyboard with 104 keys. Prior to each test, we generate a random keystroke script using Python. Each data collection session lasts 25 seconds on average and involves about 32 keystrokes, with each keystroke action lasting approximately 0.78 seconds. In total, there are 60 records collected. To ensure precise recording of both keystroke categories and timings, video recording is employed during data collection. Subsequently, a Python script is utilized to play the recorded videos frame by frame, enabling accurate recording of keystroke details. Due to the high cost and effort associated with labeling, we have annotated only *the digit keys 0-9*. Furthermore, we synchronize the sound signals emitted by the WiFi signal transmitter's speaker with the keystroke actions, aiding in aligning the recorded keystrokes with the CSI timestamp data collected.

The raw data has the shape of $[25000 \times 3 \times 30]$, with a sampling frequency of 1,000 Hz and a recording time of 25 seconds for 3 antennas and 30 subcarriers. In order to be amenable to further processing, the raw data are further

8. https://dhalperi.github.io/linux-80211n-csitool/

TABLE 3
Main Results: Performance comparison with baselines on four evaluation datasets, measured by mAP at various IoU thresholds $[0.3 : 0.1 : 0.7]$. The **best** and the second best results are marked.

Models	Backbone	Complexity	Parameters	IoU	Datasets			
					DeepSeg	*WiBehave*	*WiKeystroke*	*SeBehave*
Coarse-Fine [32]	X3D	31.64 GMac	6.0 M	-	0.36	0.23	0.46	0.82
AFSD [8]	I3D	288.77 GMac	64.41 M	0.3	0.95	0.76	0.78	0.97
				0.4	0.94	0.74	0.69	0.96
				0.5	0.92	0.69	0.63	0.94
				0.6	0.92	0.64	0.54	0.93
				0.7	0.90	0.48	0.46	0.90
				Avg	0.93	0.66	0.62	0.94
TadTR [53]	DETR	39.83 GMac	49.09 M	0.3	0.67	0.56	0.44	0.93
				0.4	0.66	0.55	0.41	0.92
				0.5	0.64	0.51	0.38	0.91
				0.6	0.59	0.46	0.27	0.89
				0.7	0.43	0.30	0.19	0.87
				Avg	0.60	0.48	0.34	0.90
TALLFormer [34]	VideoSwin	84.74 GMac	119.01 M	0.3	0.92	0.52	0.36	0.56
				0.4	0.92	0.48	0.36	0.55
				0.5	0.91	0.43	0.30	0.53
				0.6	0.90	0.38	0.28	0.51
				0.7	0.89	0.29	0.26	0.46
				Avg	0.91	0.42	0.31	0.52
STADe (Ours)	ATSE+I3D	49.13 GMac	44.29 M	0.3	0.99	0.84	0.87	0.99
				0.4	0.98	0.84	0.76	0.98
				0.5	0.98	0.82	0.70	0.98
				0.6	0.98	0.79	0.64	0.96
				0.7	0.97	0.54	0.54	0.94
				Avg	**0.98**	**0.77**	**0.69**	**0.97**

divided into a set of 8-second records. With necessary trimming, there are 174 data records included in the dataset. To make the dimensionality amenable to subsequent processes, we average the values for the three antennas to transform each record into a matrix of shape $[8000 \times 30]$.

4.3 WiBehave Dataset

WiBehave is a self-collected dataset for localizing and recognizing human activities using WiFi signals. We develop a prototype system to collect data in a daily scenario. The layout of the data collecting scenario is shown in Fig. 5. The room is 15×10 meters in size, with furniture such as closets. The hardware equipment consists of a laptop equipped with an Intel 5300 NIC, a desktop computer, and antennas. The laptop serves as the transmitter, while the desktop computer acts as the receiver. Both the transmitter and the receiver have three commercial 5dB omnidirectional antennas, each of which has 30 subcarriers with a sampling frequency of 500 Hz. The transmitter sends WiFi signals through antennas, and the receiver collects CSI data using the CSI-Tool software package installed on the Ubuntu 12.04 system. The antennas of the transmitter and receiver are both placed at a height of approximately 1.5m using tripods, with around 2.5m apart from each other.

The participants are two 22-year-old volunteers and an observer. The activity protocol consists of 7 basic activities: *walking*, *running*, *jumping*, *waving*, *bending*, *sitting*, and *standing up*. During each test, each volunteer performs 3-4 activities randomly chosen from the activity protocol within 17 seconds, and the observer manually marks each activity's start and end timestamps using a stopwatch. In total, there are 500 records collected, with each volunteer contributing 250 records.

The raw data has the shape of $[8500 \times 3 \times 30]$, with a sampling frequency of 500 Hz and a recording time of 17 seconds for 3 antennas and 30 subcarriers. To make the dimensionality amenable to subsequent processes, we average the values for the three antennas to transform each record into a matrix of shape $[8500 \times 30]$.

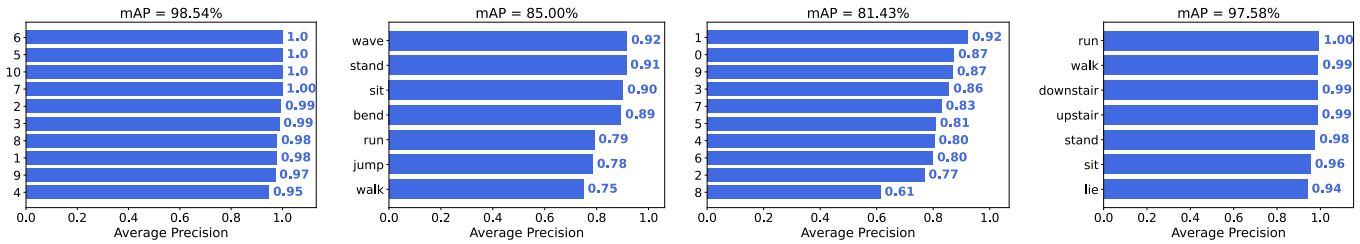
5 Experiments

5.1 Experimental Settings

Before presenting the evaluation results, we first introduce the settings used in our experiments.

Baselines. We compare with four state-of-the-art models: Coarse-Fine [32], a one-stage TAD model that employs an X3D [10] backbone; AFSD [8], which is an anchor-free TAD model with an I3D backbone; TadTR [53], an action-based TAD model that utilizes a DETR [24] backbone; and TALLFormer [34], an anchor-free TAD model that uses transformers for encoding short-term actions on top of VedioSwin [54] features and incorporates a long memory module to capture longer-duration actions. The hyper-parameters for each baseline model are finely tuned on the four sensory datasets using a dedicated validation set.

Implementation details. In our experimental setup, we employ the Adam optimizer [55] with an initial learning rate of 1e-4 and weight decay of 1e-3. The training process varies for each dataset. For *WiBehave*, we train the model for 90 epochs, utilizing a batch size of 4 and a loss balance factor α of 6. Similarly, *SeBehave* is trained for 50 epochs with a batch size of 4, while the loss balance factor α is set



(a) DeepSeg (b) WiBehave (c) WiKeystroke (d) SeBehave

Fig. 6. Categorical analyses on four datasets.

to 3.3. *DeepSeg* is trained for 60 epochs with a batch size of 2, and the loss balance factor α is 6. As for *WiKeystroke*, we conduct training for 160 epochs with a batch size of 2 and a loss balance factor α of 3.3. Unless otherwise specified, during the testing phase, we evaluate all datasets using an Intersection over Union (IoU) threshold of 0.5 for non-maximum suppression (NMS). This threshold ensures consistent and reliable measurements across the different datasets.

For I3D backbone, we did not utilize the pretrained weights. Our tests showed that I3D trained from scratch and I3D initialized with pretrained weights from video datasets performed similarly on our sensor data. This suggests that the key factor driving performance in our case is the temporal-spatial feature extraction capabilities of the model itself, rather than the transfer of visual information from video data. As a result, all of our experiments were conducted with I3D trained from scratch on the sensory dataset. This ensures that the improvements we observed are directly due to the model's ability to learn temporal and spatial patterns from sensor data, without relying on visual features from video datasets.

Evaluation metrics. Following the common practice in TAD, we adopt the mean Average Precision (mAP) as the main evaluation metric. The default IoU thresholds are $[0.3 : 0.1 : 0.7]$ for all datasets and models, except for Coarse-Fine which employs frame-wise evaluation instead of relying on IoU thresholds.

5.2 Main Results

In Table 3, we compare our model with the state-of-the-art models (*i.e.*, Coarse-Fine [32], AFSD [8], TadTR [53], and TALLFormer [34]) on the four evaluation datasets. Performance is evaluated by reporting the mAP at various IoU thresholds, ranging from 0.3 to 0.7, as well as the average mAP across all thresholds. From Table 3, we can see our model consistently outperforms all baselines, setting new state-of-the-art results on these datasets. On average, our model excels the second-best model, AFSD, by a substantial margin. We observe a 6.5 absolute percentage point improvement and a 9.13% average percentage increase in mAP@Avg over AFSD. In certain conditions, such as the mAP@0.6 on *WiBehave*, the gap even extends to 15 absolute percentage points. Given the similar framework shared with AFSD, *i.e.*, I3D backboned anchor-free model, the exceptional performance standouts the effectiveness of our contributions, namely, aligned temporal-spectral encoding and cross-cascade predictor.

The WiFi CSI datasets, *WiBehave* and *WiKeystroke*, seem more challenging than others[9], with all models performing below 0.77 mAP. We ascribe this to the complexity of CSI signals. These signals manifest intricate and obscure *passive* patterns compared to *active* three-axis accelerometer or gyroscope patterns in *SeBehave*, demanding greater efforts in feature extraction. Additionally, CSI signals are susceptible to echo delay due to obstacles, resulting in overlapped and blurred boundaries that further exacerbate the situation. On *WiBehave*, our model still outperforms all baselines with significant margins, with a remarkable improvement of 16.7% in mAP@Avg over the second-best model AFSD, which demonstrates the high effectiveness of our ATSE in extracting intricate sensory patterns and our cross-cascade predictor in modeling various dependencies.

Among all models, we find that Coarse-Fine, TadTR, and TALLFormer fall considerably short compared to other models, despite their commendable performance in video TAD. We attribute this to a number of factors. First, Coarse-Fine's coarse-stream and TALLFormer's clip sampling heavily downsample frames, resulting in the loss of informative features pertinent to fine-grained actions such as keystroking. Second, TadTR is characterized by high model complexity, employing a stack of six transformer layers. This increases the risk of overfitting on smaller datasets[10], particularly for *WiKeystroke* which has a mere 488 action instances in the training set. Third, the long-term memory mechanism in TALLFormer hinges greatly on high-quality pre-trained features. Due to the scarcity of sensory corpus data, the capabilities of long-term memory are notably constrained. Lastly, the lack of spectral treatment of action scale or resolution in TadTR and TALLFormer is also an issue, particularly given the varying sampling rates of sensory signals. The superior performance of our model when juxtaposed with these baseline models attests to its effectiveness.

5.3 Categorical Performance

Fig. 6 presents the categorical performance of our STADe model on four evaluation datasets. STADe achieves consistently good performance across all action categories.

9. The *DeepSeg* dataset appears less challenging for AFSD and STADe, with both models surpassing 0.93 mAP. This can be attributed to the clear pause intervals between activities, which makes proposals easy to detect.

10. With the exception of *SeBehave*, all other datasets contain fewer than 2,000 action instances.

TABLE 4
Cross-cascade Predictor: Effect of the three dependencies, measured by mAP@Avg for IoU thresholds of $[0.3 : 0.1 : 0.7]$.

Variant	*DeepSeg*		*WiBehave*		*WiKeystroke*		*SeBehave*	
	mAP	Δ	mAP	Δ	mAP	Δ	mAP	Δ
Full	**0.98**		**0.77**		**0.69**		**0.97**	
- temporal dependency	0.97	0.01	0.75	0.02	0.67	0.02	0.95	0.02
- category→proposal dependency	0.97	0.01	0.74	0.03	0.66	0.03	0.96	0.01
- proposal→category dependency	0.97	0.01	0.73	0.04	0.67	0.02	0.96	0.01

On the *DeepSeg* and *SeBehave* datasets, the categorical performance discrepancy remains relatively small, ranging from 0.94 to 1.0. This can be attributed to the clear pause intervals between activities and the discriminative patterns captured by smartphone sensors in these datasets. In the case of the challenging *WiBehave* dataset, the categorical performance is generally consistent, averaging around 0.85. However, there is a slight decline for the *walking*, *jumping*, and *running* categories, ranging from 0.75 to 0.79. We believe that this decline is due to the speed-related nature of these activities, which makes it less discriminative to distinguish between *walking* and *running*. The relatively low score for the "jumping" action, *i.e.*, average precision of 0.78, can be attributed to the action's nature. Unlike more continuous and gradual motions, "jumping" is a discrete action characterized by a sudden and brief change in position, which can be harder to capture accurately in sensory data. For the *WiKeystroke* dataset, we observe a performance drop for the number keys "2" and "8" compared to other keys, with scores ranging from 0.61 to 0.77, while the average performance is 0.81. This drop in performance can be attributed to the positioning of these keys in the middle of using either hand, which makes their patterns less discriminative.

Another observation is the difference in performance among different datasets. For example, the mAP scores for *walking* and *running* are lower in the *WiBehave* dataset, compared to the *SeBehave* dataset. This is mainly due to the *sensor modality differences* (WiFi vs. accelerometers/gyroscopes). The *SeBehave* dataset uses smartphone sensors (accelerometers and gyroscopes) that *directly capture motion dynamics*. The high sampling rate of the embedded sensors (200 Hz) is sufficient to precisely track walking, running, and other activities. In contrast, *WiBehave* relies on WiFi signals (CSI), which *indirectly sense body movement* and are often influenced by both body motion and environmental factors, making them more susceptible to noise and interference. The indirect nature of WiFi data makes it harder to distinguish between similar dynamic actions like *walking* and *running*.

Overall, the categorical analysis demonstrates that our STADe model performs equally well across different actions in various datasets. This indicates that our model effectively extracts action patterns and avoids relying solely on biases associated with a few specific actions.

5.4 Ablation Study

5.4.1 Cross-cascade Predictor

To illustrate the effectiveness of each component in our cross-cascade predictor, we conduct an ablation study, where we individually remove each of the three dependencies (via removing corresponding features) and measure the effect on the mAP@Avg for IoU thresholds from 0.3 to 0.7. The results are presented in Table 4. The full model achieves the highest mAP@Avg on all datasets. Removing the temporal dependency results in an mAP decrease on all datasets (average 0.018), suggesting that sequential order aids in action identification and mitigates sensory signal delays. The removal of the category→proposal dependency causes a 0.02-point performance drop on average, corroborating the TAD practice of generating action categories based on top proposals. Interestingly, removing the proposal→category dependency also leads to an identical 0.02-point performance decline. This solidifies the significance of the bidirectional dependency between action proposals and categories in improving the performance of the model. The results of this ablation study provide empirical evidence to our observations regarding the temporal and proposal→category dependencies, manifesting their instrumental contribution to the final sensory TAD results.

5.4.2 Various Choices of the Feature Encoder

In Table 5, we evaluated the impact of using different feature encoders (*i.e.*, backbones) on sensory TAD tasks. We report mAP scores on the *WiKeystroke* dataset with an IoU threshold of 0.3. We tested eight backbone settings, including a standard CNN containing 4 convolutional blocks (each having a convolution, a BatchNorm, and a ReLU operation), ResNet [56], C3D [40], R3D [41], R(2+1)D [42], vanilla I3D [9], and two configurations of our proposed architecture, *i.e.*, ATSE+I3D$\mathrm{parallel}$ and ATSE+I3D$\mathrm{stacking}$. In the parallel configuration, ATSE and I3D operate independently on the raw input to extract features. Conversely, in the stacking configuration, I3D is stacked on top of ATSE.

The traditional 2D networks, such as the basic CNN and ResNet, achieved mAP@0.3 scores of 0.43 and 0.52, respectively. These networks rely solely on 2D convolutions to extract spatial features from individual frames, the inability of these models to model temporal variations adequately results in their lower performance. In contrast, the 3D convolutional networks show a significant improvement in performance due to their capability to extract both spatial and temporal features. The C3D model, which uses full 3D convolutions to process the data, achieved mAP score at 0.8, indicating its strong ability to integrate spatiotemporal information. R3D and I3D, which also leverage 3D convolutions (with I3D specifically utilizing inflated 2D kernels from pre-trained models), both reached scores of 0.81. On the other hand, the R(2+1)D model, which decomposes the 3D convolution into a 2D spatial and a 1D temporal convolution, achieved the score of 0.79. This slight drop in performance may be due to its separation of spatial and

TABLE 5
Backbone: Effect of using different feature encoder on the *WiKeystroke* dataset, measured by mAP@0.3.

CNN	ResNet	C3D	R3D	R(2+1)D	I3D	ATSE+I3D$_{\text{parallel}}$	ATSE+I3D$_{\text{stacking}}$
0.43	0.52	0.80	0.81	0.79	0.81	0.82	0.84

TABLE 6
Proportion of Training Data Size: mAP at different IoU thresholds in $[0.3 : 0.1 : 0.7]$ for various proportion of training data size on the *WiBehave* dataset.

Training data used	mAP					
	Iou=0.3	Iou=0.4	Iou=0.5	Iou=0.6	Iou=0.7	Avg
20%	0.53	0.52	0.47	0.38	0.26	0.43
40%	0.69	0.69	0.66	0.41	0.29	0.55
60%	0.76	0.74	0.66	0.47	0.32	0.59
80%	0.81	0.79	0.72	0.66	0.47	0.69
100%	0.84	0.84	0.82	0.79	0.54	0.77

temporal processing, which, while reducing the number of parameters and computational cost, might not capture the interdependencies between these dimensions as effectively in this specific sensor data context. Among the 3D models, the differences among C3D, R3D, and I3D are relatively small. Finally, the integration of the ATSE module, especially through stacking, enhances feature representations to achieve the best overall performance.

5.4.3 Training Data Size

Table 6 presents the impact of different proportions of training data size. We randomly sample the required proportion from the full training set for different scales, ensuring the subsets are unbiased and representative of the overall training data distribution. We report the mAP scores under different training proportions, ranging from 20% to 100%. The results suggest the proportion of training data size significantly influences the model's performance. As the proportion of training data increases, there is a general trend of improved mAP scores across all IoU thresholds. At a training data size of 20%, the average mAP score is 0.43, indicating relatively moderate performance. However, as the proportion of training data size increases to 40%, 60%, 80%, and 100%, the mAP scores show noticeable improvements. Particularly, at 100% training data size, the highest mAP score is achieved at 0.77. This suggests that utilizing the entire training dataset leads to superior performance in action detection. For the sensory TAD task, allocating a larger proportion of the training dataset improves the model's ability to learn and generalize, resulting in enhanced action detection performance.

5.5 Parameter Sensitivity Analysis

In order to evaluate the influence of hyper-parameters on STADe's mAP and offer an empirical method for their tuning, we perform parameter sensitivity analyses. These analyses are conducted to examine the model's performance across different resolution schemes, proposal pooling sizes, and balance factors.

5.5.1 Various Resolution Schemes

Table 7 shows the impact of different numbers of resolutions on the *WiBehave* dataset. We select five different resolution schemes, ranging from more fine-grained to coarser resolutions. Each resolution scheme consists of specific resolutions or scales employed during the model's evaluation. The mAP scores and computational time costs are reported for IoU in $[0.3 : 0.1 : 0.7]$.

The results in Table 7 reveal that the choice of different numbers of resolutions has a noticeable impact on the model's performance. On average, the finest scheme, {5, 7, 16, 32, 48, 64, 80, 96, 112, 128}, achieves the highest mAP score of 0.77, indicating its effectiveness in capturing precise action boundaries. This highlights the potential benefits of employing finer schemes that offer more options for automatically determining suitable spectral window sizes. As the resolution scheme becomes coarser, the average mAP decreases. The scheme with only two resolutions, *i.e.*, {5, 80}, exhibits the lowest mAP values of 0.64. This decline in performance for fewer resolutions can be attributed to the Fourier uncertainty principle, which affects the ability to precisely locate events in the time-frequency domain.

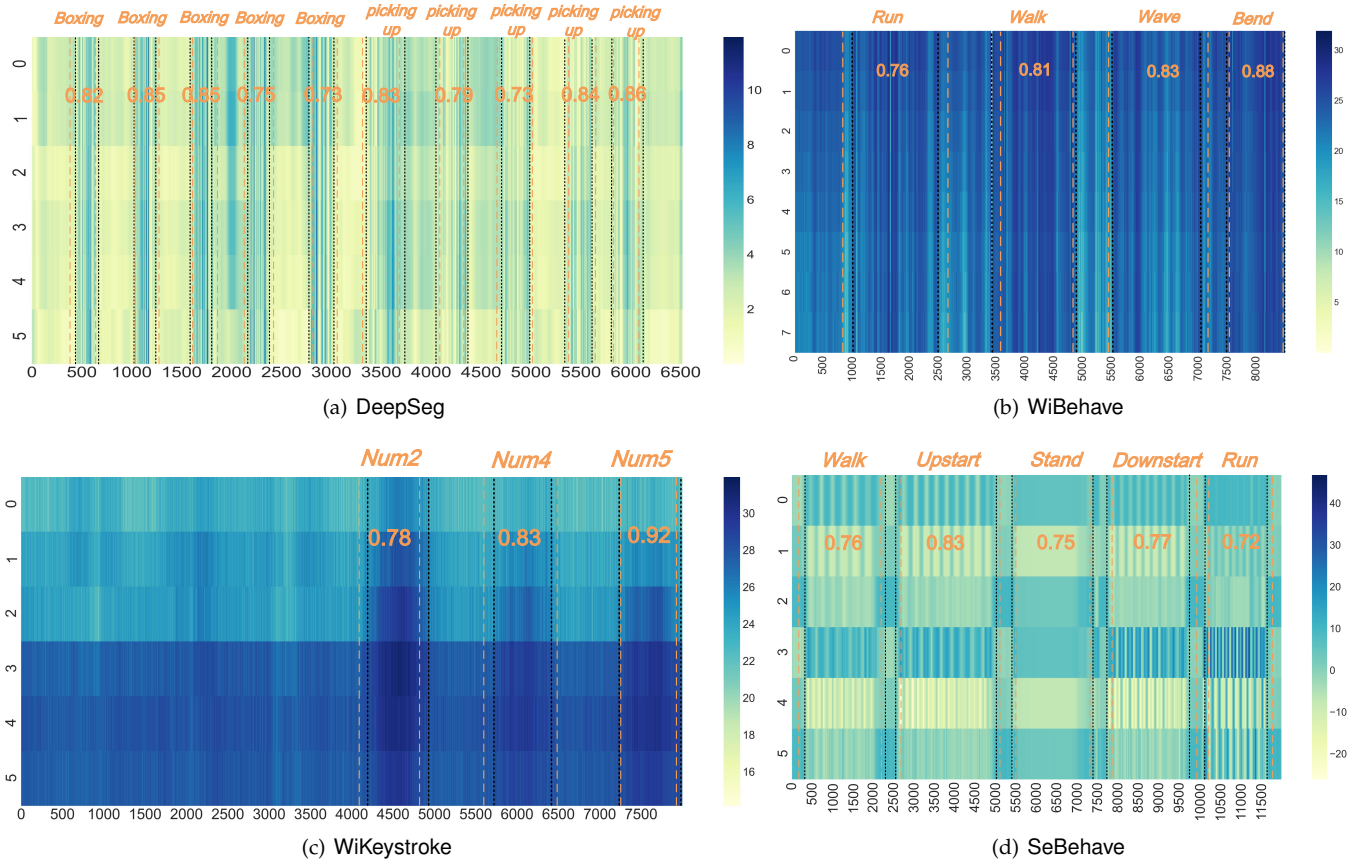
However, it is worth noting that the fine scheme substantially increases computational and model complexity since each resolution requires an additional ATSE module. Table 7 show that the training time decreases as fewer resolutions are used. For instance, the {5,7,16,32,48,64,80,96,112,128} scheme requires 247 seconds for training, while the {5,80} scheme, with the fewest resolutions, only requires 81 seconds. This indicates that reducing the number of resolutions accelerates training. The inference time remains relatively stable across resolution schemes, consistently ranging from 5 to 7 seconds. The inference speed (measured in time series per second) increases as fewer resolutions are used. The {5,80} scheme achieves the highest inference speed at 30 ts/s, significantly outperforming the {5,7,16,32,48,64,80,96,112,128} scheme, which achieves only 21.42 ts/s.

Overall, striking a balance between capturing details and maintaining computational efficiency becomes essential in practical implementations. By carefully selecting a resolution scheme that strikes a balance between these factors, we can achieve satisfactory results in capturing meaningful action information while maintaining acceptable computational costs.

Table 8 shows the performance under various granularity of resolutions. In this setting, each scheme is fixed to having 6 resolutions, while each resolution may have a different granularity, *i.e.*, the size of the kernel. Among the evaluated resolution schemes, the scheme {5, 7, 32, 56, 72, 112} with minimal average kernel sizes achieves an average mAP score of 0.71. As the average length of the resolutions increases, such as in the schemes {5, 9, 34, 58, 74, 114} and

TABLE 7
Resolution Scheme: mAP and computational time costs at different IoU thresholds in $[0.3 : 0.1 : 0.7]$ for various numbers of resolutions on the *WiBehave* dataset.

Resolution Scheme	mAP						Time Cost		
	Iou=0.3	Iou=0.4	Iou=0.5	Iou=0.6	Iou=0.7	Avg	Training	Inference	Infer Speed
{5,7,16,32,48,64,80,96,112,128}	0.84	0.84	0.82	0.79	0.54	0.77	247s	7s	21.42 ts/s
{5,7,16,48,64,80,112,128}	0.82	0.82	0.78	0.72	0.50	0.73	204s	6s	25 ts/s
{5,16,48,64,80,128}	0.84	0.83	0.81	0.70	0.37	0.71	163s	6s	25 ts/s
{5,48,80,128}	0.80	0.78	0.77	0.67	0.44	0.69	122s	6s	25 ts/s
{5,80}	0.75	0.74	0.71	0.62	0.40	0.64	81s	5s	30 ts/s



(a) DeepSeg (b) WiBehave

(c) WiKeystroke (d) SeBehave

Fig. 7. **Visualization**: Spectral maps generated by averaging sensory channel values from one example in each of the four datasets. The ground-truth and predicted proposals are represented by a pair of black vertical lines and orange vertical lines, respectively. The IoU values of the predicted proposals are also indicated. The correct categories are indicated at the top of each figure.

TABLE 8
Resolution Scheme: mAP at different IoU thresholds in $[0.3 : 0.1 : 0.7]$ for various granularity of resolutions on the *WiBehave* dataset.

Resolution Scheme	mAP					
	Iou=0.3	Iou=0.4	Iou=0.5	Iou=0.6	Iou=0.7	Avg
{5,7 32,56,72,112}	0.84	0.83	0.81	0.7	0.37	0.71
{5,9,34,58,74,114}	0.81	0.81	0.76	0.71	0.51	0.72
{5,16,48,64,80,128}	0.86	0.85	0.81	0.75	0.51	0.76
{5,24,56,72,88,136}	0.87	0.84	0.83	0.76	0.53	0.77

TABLE 9
Proposal Pooling Size: mAP at different IoU thresholds in $[0.3 : 0.1 : 0.7]$ for various proposal pooling size on the *WiBehave* dataset.

Number of proposals	mAP					
	Iou=0.3	Iou=0.4	Iou=0.5	Iou=0.6	Iou=0.7	Avg
42	0.34	0.32	0.28	0.23	0.17	0.27
88	0.85	0.84	0.80	0.69	0.44	0.72
178	0.84	0.84	0.82	0.79	0.54	0.77
360	0.89	0.88	0.86	0.76	0.58	0.79

$\{5, 16, 48, 64, 80, 128\}$, higher performances are achieved, with an average mAP of 0.72 and 0.76, respectively. Notably, the scheme $\{5, 24, 56, 72, 88, 136\}$ achieves the highest average mAP of 0.77, consistently performing well across various IoU thresholds. The results suggest the significance of resolution granularity in the task of TAD. Selecting an appropriate set of resolutions has an impact on the model's performance, as observed in the improved mAP scores with larger average lengths of resolutions.

TABLE 10
Balance Factor: mAP at different IoU thresholds in $[0.3 : 0.1 : 0.7]$ for various balance factors on the *WiBehave* dataset.

Balance factor α	mAP					
	Iou=0.3	Iou=0.4	Iou=0.5	Iou=0.6	Iou=0.7	Avg
1	0.23	0.20	0.15	0.08	0.04	0.14
6	0.84	0.84	0.82	0.79	0.54	0.77
10	0.46	0.45	0.39	0.32	0.23	0.37

5.5.2 Various Proposal Pooling Sizes

Table 9 illustrates the influence of various proposal pooling sizes on the *WiBehave* dataset. The mAP scores are reported for proposal pooling sizes ranging from 42 to 360. The proposal pooling size directly impacts the granularity and coverage of intermediate proposals generated during the proposal prediction module before non-maximum suppression.

As can be observed from Table 9, when the proposal size is set to a relatively small value such as 42, the resulting mAP score of 0.27 is notably low. This is not surprising since this limited number of proposals may not adequately cover the wide range of variations present in action instances. As a result, the detection performance is compromised. However, as the proposal size increases to 88, a drastic performance improvement is observed, with the mAP soaring from 0.27 to 0.72. This substantial increase highlights the importance of having a sufficient number of anchors to accurately capture the diverse range of action instances. Furthermore, with further increments in the number of anchors, such as in the cases of 178 and 360, the mAP scores continue to rise, albeit at a slightly slower rate compared to the scheme with 88 proposals. The scheme with 360 proposals attains the highest mAP scores across all IoU thresholds, indicating its effectiveness in accurately localizing actions.

These results emphasize the significance of selecting a reasonable number of anchors as a crucial hyper-parameter. A balance needs to ensure the accurate detection of action instances while avoiding an excessive increase in computational complexity. By appropriately adjusting the proposal size, the model can effectively capture diverse action instances and achieve improved performance in action detection tasks.

5.5.3 Various Balance Factors

Table 10 presents the impact of various balance factors α (Eq. 13) for the category classification and location regression losses. The results indicate that the model's performance is quite sensitive to the choice of the balance factor. When α is set to 1, indicating equal weights for classification and regression, the average mAP score is remarkably low at only 0.14. This suggests that equal emphasis on both sub-tasks at the loss level is insufficient for sensory TAD. A balance factor of 6 exhibits a drastic improvement over equal weights, achieving the highest average mAP score of 0.77, approximately 5.5 times better than that with $\alpha = 1$. This indicates that emphasizing the location regression sub-task more than category classification benefits overall sensory TAD performance. This finding aligns with the intuition that accurate category prediction relies on correct proposal localization. Interestingly, when the emphasis on regression is further increased, *e.g.*, with a balance factor of 10, the mAP drops to 0.37. Although performance improves compared to the balance factor of 1, it fails to match the performance achieved with the balance factor of 6. The above findings show the importance of carefully selecting an appropriate balance factor during model training. Placing emphasis on location regression, with an appropriate intensity as demonstrated by the balance factor of 6, effectively balances the two sub-tasks and yields superior detection results.

5.6 Visualization

In Fig. 7, we present spectral maps generated by averaging sensory channel values from one example in each of the four datasets. The ground-truth and predicted proposals are represented by black and orange vertical lines, respectively. The IoU values of the predicted proposals are also indicated. The correct categories are shown at the top of each figure.

As expected, the data exhibits meaningful spectral patterns during the acting period, particularly for the sample from the *WiKeystroke* dataset (Fig. 7(c)), where the action period shows a higher amplitude compared to the non-acting range. This highlights the importance of including the spectral component in our ATSE encoding. We also observe that the patterns in the four datasets vary. For instance, the amplitudes of the two WiFi CSI locomotion samples, Fig. 7(a) and Fig. 7(b), exhibit fine and drastic changes over time. In contrast, the amplitude of *WiKeystroke* (Fig. 7(c)) is smoother, while the record from *SeBehave* (Fig. 7(d)) shows more discrete patterns. These differences can be attributed to variations in sensor types, sampling rates, and action types. Nevertheless, our model demonstrates high generality and adaptability among the four cases from different datasets. Furthermore, our model is also adaptive to different distributions of actions. Fig. 7(a) depicts a scenario with high-density actions, while Fig. 7(c) shows sparse actions with the first half of the records containing non-effective actions. Despite these variations, our model correctly identifies the proposals with high IoU scores and associates them with the respective categories. This demonstrates the high effectiveness of our model, STADe, and its interpretability.

6 Conclusion

While deep neural models have established their dominance in video TAD, their application to sensory signals presents distinct challenges stemming from varying sampling rates, intricate pattern structures, and subtle, noise-prone patterns. In response to these obstacles, we introduce STADe, a specialized model designed for sensory TAD. STADe leverages aligned temporal-spectral encoding to adapt feature representations to both temporal and spectral patterns, employing deep fusion to accommodate multi-resolution and multi-scale patterns associated with different sampling rates. Additionally, we propose an innovative cross-cascade predictor for proposals and categories, addressing dependencies overlooked by existing methods. Future research directions include enhancing TAD in sensory signals by tackling these challenges, thereby unlocking its potential in a wide array of application domains. Furthermore, we

create three novel datasets for sensory TAD using various sensors. These datasets exhibit diverse sensor types, action categories, and sampling rates, facilitating comprehensive evaluations of sensory TAD methodologies. We believe that the release of these datasets will significantly contribute to future research in sensory TAD.

REFERENCES

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

[2] C. Xiao, Y. Lei, Y. Ma, F. Zhou, and Z. Qin, "DeepSeg: Deep-learning-based activity segmentation framework for activity recognition using WiFi," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5669–5681, 2020.

[3] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3.0: Zero-effort cross-domain gesture recognition with Wi-Fi," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8671–8688, 2021.

[4] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[5] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[6] L. L. Bello and W. Steiner, "A perspective on IEEE time-sensitive networking for industrial communication and automation systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019.

[7] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2914–2923.

[8] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Learning salient boundary feature for anchor-free temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3320–3329.

[9] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[10] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 203–213.

[11] S. Aminikhanghahi, T. Wang, and D. J. Cook, "Real-time change point detection with application to smart home time series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 1010–1023, 2018.

[12] B. Li, W. Cui, L. Zhang, C. Zhu, W. Wang, I. Tsang, and J. T. Zhou, "DifFormer: Multi-resolutional differencing transformer with dynamic ranging for time series analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[13] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 286–293.

[14] H. Xu, A. Das, and K. Saenko, "R-C3D: Region convolutional 3D network for temporal activity detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5783–5792.

[15] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the ACM International Conference on Multimedia*, 2017, pp. 988–996.

[16] G. B. Folland and A. Sitaram, "The uncertainty principle: A mathematical survey," *Journal of Fourier Analysis and Applications*, vol. 3, pp. 207–238, 1997.

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

[18] Kuo, *Active Noise Control Systems: Algorithms and DSP Implementations*. Wiley, 1996.

[19] S. Satkin and M. Hebert, "Modeling the temporal extent of actions," in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 536–548.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, pp. 60–79, 2013.

[21] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[23] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[24] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proceedings of the International Conference on Learning Representations*, 2021.

[25] H. Kuehne, A. Richard, and J. Gall, "A hybrid RNN-HMM approach for weakly supervised temporal action segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 765–779, 2018.

[26] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[27] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang, "A pursuit of temporal accuracy in general activity detection," *arXiv preprint arXiv:1703.02716*, 2017.

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

[29] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "TURN TAP: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3628–3636.

[30] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 344–353.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[32] K. Kahatapitiya and M. S. Ryoo, "Coarse-fine networks for temporal activity detection in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8385–8394.

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

[34] F. Cheng and G. Bertasius, "TALLFormer: Temporal action localization with a long-memory transformer," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 503–521.

[35] Y. Liu, Z. Lu, J. Li, T. Yang, and C. Yao, "Deep image-to-video adaptation and fusion networks for action recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 3168–3182, 2019.

[36] Y. Liu, K. Wang, G. Li, and L. Lin, "Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 5573–5588, 2021.

[37] Y. Liu, K. Wang, L. Liu, H. Lan, and L. Lin, "Tcgl: Temporal contrastive graph for self-supervised video representation learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 1978–1993, 2022.

[38] H. Yan, Y. Liu, Y. Wei, Z. Li, G. Li, and L. Lin, "Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5606–5618.

[39] S. Li, R. R. Chowdhury, J. Shang, R. K. Gupta, and D. Hong, "Units: Short-time fourier inspired neural networks for sensory time series classification," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 234–247.

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

[41] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3d residual networks for action recognition," in

Proceedings of the IEEE international conference on computer vision workshops, 2017, pp. 3154–3160.
[42] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.
[43] S. L. Lauritzen, *Graphical models*. Clarendon Press, 1996, vol. 17.
[44] M. H. Kalos and P. A. Whitlock, *Monte carlo methods*. John Wiley & Sons, 2009.
[45] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
[46] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.
[47] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.
[48] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9759–9768.
[49] M. G. Sohrab and M. Miwa, "Deep exhaustive model for nested named entity recognition," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2843–2849.
[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
[51] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS – improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5561–5569.
[52] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie, "Human respiration detection with commodity WiFi devices: Do user location and body orientation matter?" in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 25–36.
[53] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Zhang, S. Bai, and X. Bai, "End-to-end temporal action detection with transformer," *IEEE Transactions on Image Processing*, vol. 31, pp. 5427–5441, 2022.
[54] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3202–3211.
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.