OnPoint: Offline-to-Online Multi-Level Distillation for Point-Supervised Online Temporal Action Localization

Sakib Reza[1,3⋆], Gauri Jagatap[2], Mohsen Moghaddam[3], Octavia Camps[1], and Andrea Fanelli[2]

[1] Northeastern University, Boston, MA 02115, USA
{reza.s, o.camps}@northeastern.edu
[2] Dolby Laboratories, Inc., San Francisco, CA 94103, USA
{gauri.jagatap, andrea.fanelli}@dolby.com
[3] Georgia Institute of Technology, Atlanta, GA 30332, USA
{sreza32, mohsen.moghaddam}@gatech.edu

Abstract. Temporal Action Localization (TAL) typically relies on segment annotations or offline access to full videos, limiting scalability and online use. We introduce Point-Supervised Online TAL (POTAL), which localizes actions in streaming videos using only one temporal point per instance. To solve POTAL, we propose OnPoint, an offline-to-online multi-level distillation framework that transfers knowledge from a point-supervised offline teacher to an online student via (i) pseudo-segment instance distillation, (ii) class-activation sequence distillation, and (iii) anticipatory window-level distillation. We further improve robustness by incorporating the original point labels into student training and by refining anchor decoding with actionness-guided attention calibration. Experiments on five datasets show OnPoint consistently outperforms strong baselines, establishing a solid foundation for POTAL. [†]

Keywords: Video Understanding · Temporal Action Localization

1 Introduction

With the rapid growth of video platforms, Temporal Action Localization (TAL) [39, 41], which identifies action boundaries and class labels in untrimmed videos, has become a central task in computer vision. However, many real-world systems must localize actions directly from *streaming video* while minimizing annotation cost. Examples include augmented reality (AR) task assistance for surgical training or industrial maintenance [1, 47], as well as live sports analytics [34], surveillance [38], and robotics [7]. In such settings, models must make frame-by-frame predictions without future context while operating under limited supervision, as dense action boundary annotation is often costly and infeasible at scale.

⋆ Work primarily done during an internship at Dolby Laboratories.
† Project Page: https://sakibreza.github.io/OnPoint/

A key challenge arises from the *inference regime*. Most TAL methods [5, 17, 18, 40, 42–44, 46, 48] assume offline access to complete videos, allowing models to leverage future context during training and inference. In contrast, real deployments require *online inference*, where actions must be localized in streaming video as frames arrive. Recent Online Temporal Action Localization (OnTAL) approaches [9, 31] address this setting but rely on full segment-level supervision during training, limiting scalability in continuously recorded environments where annotation is scarce.

A second challenge concerns the *supervision cost*. Training state-of-the-art TAL or OnTAL models typically requires dense temporal annotations specifying action start and end boundaries, which are expensive to obtain. Recent large multimodal foundation models [2, 25, 29] show promising zero-shot capabilities but remain unreliable for precise temporal action instance detection and boundary localization (Fig. 1). Consequently, fully automatic label generation remains impractical for many real-world deployments.

Fig. 1: Labeling example. Ground truth, human point labels, pseudo segments from our point-supervised offline TAL teacher, and Gemini 2.5 Flash [2] MLLM labels. Gemini often merges instances and misplaces boundaries (29.2% avg mAP), while our teacher yields accurate pseudo labels (90.3%) on THUMOS [6], underscoring the value of weak human annotation. (Sec. C)

A practical alternative is *point supervision*, where each action instance is annotated with a single timestamp. Prior work [24] and our supplementary study (Supp. Sec. C) show that point labeling can reduce annotation effort by up to 6× while maintaining reliability. This paradigm has been explored in Point-Supervised TAL (PS-TAL) [24, 49], which learns action boundaries from sparse point labels but assumes offline access to the full video.

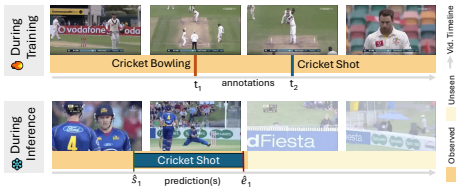

Fig. 2: POTAL task. Training uses one timestamp per action instance. At test time, the model outputs action class and boundaries online, emitting each segment immediately when the action ends (no future frames).

Despite substantial progress in PS-TAL and OnTAL, no prior work addresses their intersection. PS-TAL methods rely on full video context, violating the causality constraints required for streaming inference, while existing OnTAL methods require full segment supervision. Simply combining the two paradigms collapses supervision assumptions or breaks the online setting.

To bridge this gap, we introduce **Point-Supervised Online Temporal Action Localization (POTAL)**, a new task that combines the label efficiency of point annotations with the strict streaming constraints of online inference (Fig. 2). POTAL defines a new problem setting (Sec. 3.1), establishes evaluation

benchmarks, and introduces both distillation-based and distillation-free baselines tailored to this scenario.

To tackle POTAL, we propose **OnPoint**, an offline-to-online distillation framework that uses a point-supervised offline TAL model as a teacher for a strictly online student. The teacher leverages full-video context to infer boundaries from sparse points and supplies structured supervision—pseudo segments, frame-wise activations, and future-window cues—enabling the student to localize actions under streaming constraints.

While offline-to-online distillation has been studied in tasks such as video instance segmentation [10], spatio-temporal action detection [26], language modeling [19], and reinforcement learning [14], it remains unexplored in temporal action localization (TAL). To the best of our knowledge, **OnPoint** is the first framework to distill knowledge from an offline TAL model into an online counterpart. It introduces a purpose-built multi-level distillation framework, with each component designed to address a specific challenge of the POTAL setting. Our key contributions are:

1. **A new task setting: Point-Supervised Online TAL (POTAL).** We formulate the first temporal action localization problem that simultaneously enforces strict online inference (no future frames, no offline revision) and point-level supervision (one timestamp per instance). We provide protocols, evaluation, and strong baselines to establish POTAL as a benchmark setting.
2. **OnPoint: the first offline-to-online distillation framework for TAL under point supervision. OnPoint** transfers knowledge from a point-supervised offline TAL model to a strictly online detector, directly addressing the two core challenges of POTAL: missing future context and sparse supervision. Training an online model using only point-level supervision is inherently difficult due to weak labels and the absence of future context. To address this, we introduce three complementary distillation signals:
   (a) **Providing structured boundary supervision through instance-level pseudo-label distillation.** Segment-level pseudo ground truth generated by the offline teacher supervises the anchor-based instance prediction module, providing structured boundary guidance (Sec. 3.7).
   (b) **Enhancing temporal representation learning via dense frame-level distillation.** We align the student's frame-wise predictions with the teacher's Class Activation Sequence (CAS) via Class-Activation Sub-Sequence (CASS) supervision. While not directly used for boundary regression, this dense signal strengthens temporal representations and improves backbone feature quality (Sec. 3.3).
   (c) **Compensating for missing future context through anticipatory window-level distillation.** To mitigate the absence of future context in online inference, we distill the teacher's next-window action likelihoods into the student, guiding it toward imminent action boundaries and enabling timely localization without access to future frames (Sec. 3.4).
3. **Stabilizing online training under noisy pseudo labels via direct anchor-level point supervision.** Because pseudo labels may be imperfect,

we directly incorporate ground-truth action points as an additional anchor-level supervisory signal. This weak but reliable supervision stabilizes training and mitigates error propagation from the teacher (Sec. 3.6).

4. **Improving temporal proposal decoding through actionness-guided attention calibration.** Standard anchor-based decoders rely on learned attention queries that lack explicit action priors. We introduce an actionness-guided attention calibration mechanism that leverages frame-wise actionness scores predicted by CASS to emphasize action-relevant regions and suppress background noise, improving temporal proposal accuracy (Sec. 3.5).

Extensive experiments (Sec. 4.2) on five datasets show OnPoint consistently improves over strong baselines (up to +7% mAP) and is competitive with select fully supervised online methods, establishing a strong foundation for POTAL.

2 Related Work

Temporal Action Localization (TAL) is a fully-supervised, offline framework to detect temporal boundaries and classify action instances. Existing one-stage approaches [16, 32, 33, 48] jointly localize and classify actions in a single step, while two-stage pipelines [17, 28] first generate candidate proposals and then perform classification. Despite their impressive performance, these models rely heavily on densely annotated temporal boundaries and action labels for training. Moreover, their offline inference paradigm requires access to entire video sequences, restricting their deployment in online or streaming environments.

Point-Supervised Temporal Action Localization (PS-TAL) is an *offline* weakly-supervised setting where each action instance is labeled by a single timestamp, providing a label-efficient alternative to full supervision and typically outperforming video-level supervision [50]. Existing methods fall into three lines: *pseudo-label generation* that expands points to nearby frames for dense supervision [4, 24, 27]; *sequence/proposal modeling* that uses points to improve action completeness and calibrate snippet- or instance-level confidence [13, 45, 49]; and *boundary refinement/global context* that searches for precise boundaries or leverages structured frameworks (e.g., optimal transport, DETR-style decoders) for long-range dependencies [20,21]. However, all PS-TAL methods assume full-video access and are therefore unsuitable for streaming scenarios that require causal, incremental predictions.

Online Temporal Action Localization (OnTAL), introduced in [9], aims to localize action boundaries and classify action instances in online settings, without relying on future frames or offline post-processing. Early approaches such as CAG-QIL [9] and SimOn [36] utilize per-frame action detection or online action detection backbones, followed by grouping consecutive predictions to generate action proposals. 2PESNet [12] advanced the field by introducing a two-stage framework for explicit start and end time detection at the instance level. Later, anchor-based approaches such as OAT [11] and HAT [31] were proposed to avoid temporal fragmentation and imprecise boundaries, while ActionSwitch [8] addressed the challenge of overlapping action localization. Furthermore, memory-

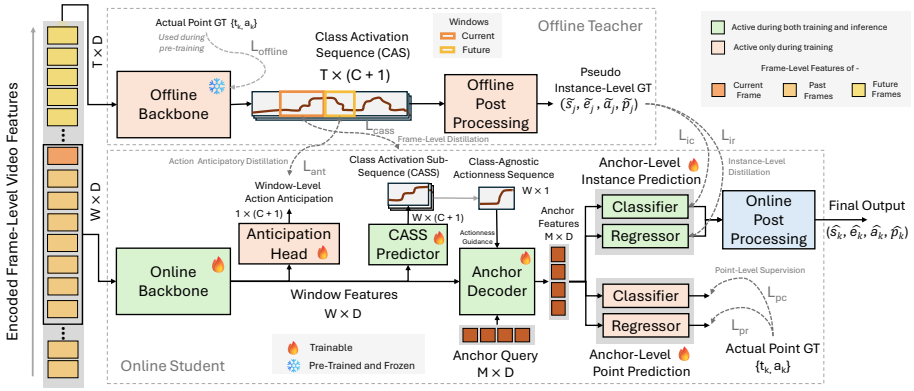

Fig. 3: An overview of our proposed OnPoint framework for POTAL task. The offline teacher model, pre-trained with point-level annotations, is frozen during training. We distill knowledge into the online student model using pseudo ground truth, frame-wise class activations, and window-level action anticipation objectives. Additionally, original point annotations are directly leveraged to supervise the online model.

augmented models like MATR [35] and HAT [31] leveraged long-term temporal dependencies to enhance instance-level reasoning. While OnTAL satisfies the online inference constraint of traditional TAL, current methods remain label-intensive, as they require full boundary annotations during training.

3 Method

3.1 Problem Definition

We introduce Point-Supervised Online Temporal Action Localization (POTAL), a novel task that aims to detect and classify action segments in streaming video using only point-level supervision, i.e., a single annotated timestamp t_k and class label a_k per action instance during training. Given a video stream $V = \{v_i\}_{i=1}^{T}$, the model observes only the partial sequence $V_{1:t} = \{v_i\}_{i=1}^{t}$ at time t and must produce action proposals $(\hat{s}, \hat{e}, \hat{a}, \hat{p})$, where $\hat{s}$ and $\hat{e}$ denote predicted start and end times, $\hat{a}$ is the action class, and $\hat{p}$ is the confidence score. At inference, the model must produce proposals immediately upon detecting action ends, with no option for revision. The goal is to sequentially recover the full action set $\Psi = \{(\hat{s}_k, \hat{e}_k, \hat{a}_k)\}_{k=1}^{\hat{K}}$ in a strictly online, label-efficient setting.

3.2 Offline-to-Online Distillation Framework

Training a standalone online model using only point-level supervision is challenging due to limited boundary cues and absence of future context. In contrast, an offline model leverages the full temporal context to infer action boundaries from sparse point annotations, generating high-quality pseudo-labels. Thus, we introduce OnPoint (Fig. 3), a multi-level distillation framework comprising two core

components: (1) a point-supervised offline teacher that generates pseudo-labels and class activation sequences (CAS), and (2) an online student that leverages these multi-level supervisory signals to perform online action localization using sliding-window inputs.

Offline Teacher. It processes the full video features $X_{\text{full}} \in \mathbb{R}^{T \times D}$, where T is the number of frames and D is the feature dimension, using an offline backbone to generate a class activation sequence (CAS) $A \in \mathbb{R}^{T \times (C+1)}$, with C denoting the number of action classes and the extra channel representing background. Offline post-processing on A yields pseudo ground truth segments $(\tilde{s}, \tilde{e}, \tilde{a}, \tilde{p})$, where $\tilde{s}$ and $\tilde{e}$ are the predicted start and end times, $\tilde{a}$ is the action label, and $\tilde{p}$ is the confidence score. The offline backbone is trained using only point-level annotations. We employ HR-Pro's backbone [49] for sparse action datasets like THUMOS'14 [6], and TSASPC's backbone [4] for dense datasets such as EGTEA [15] and HOI4D [23], given its suitability for densely annotated scenarios. We also follow the training and post-processing strategies from the mentioned prior works, with additional details provided in the supplementary material.

Online Student. It operates in a streaming fashion using a sliding window mechanism. At each time step t, it collects the current frame and the preceding $W-1$ frames to form a feature window $X_t \in \mathbb{R}^{W \times D}$, where W is the window size and D is the feature dimension. This input is passed through an online backbone built with standard Transformer encoders [37] to capture temporal dependencies, producing intermediate representations $F_t \in \mathbb{R}^{W \times D}$.

These features are then used by three key modules: (1) a Class Activation Subsequence (CASS) predictor that estimates frame-wise class scores, (2) a window-level anticipation head that predicts potential future actions, and (3) an anchor decoder that generates anchor-level features for a set of predefined temporal anchors within the window. The anchor features are further processed by classification and regression heads, followed by online post-processing to produce final action predictions.

Training signals. The training of the online model is guided by the offline teacher in a multi-level distillation manner: instance-level pseudo ground truth supervises anchor-level classification and localization; the offline CAS supervises both the anticipation head and the CASS predictor; and the original point annotations are used to supervise anchor-level point prediction. More details on each module, as well as the training and inference procedures, are provided in the sections below.

3.3 Class-Activation Sub-Sequence (CASS) Distillation

To provide dense frame-level supervision, we introduce CASS distillation, which learns to predict frame-wise action class scores within each sliding window. By aligning the online model's predictions with the offline teacher's Class Activation Sequence (CAS), CASS offers fine-grained temporal guidance that strengthens feature learning. This supervision enables the student to capture subtle action transitions more effectively, improving temporal reasoning when future context is unavailable.

Given the intermediate window-level feature representation $F_t \in \mathbb{R}^{W \times D}$, where W is the window size and D is the feature dimension, the CASS predictor maps each frame embedding to class probabilities using a lightweight two-layer feedforward network. Specifically, the network consists of two linear layers with a ReLU activation in between, projecting from D to D, and then from D to $C+1$, where C is the number of action classes and the additional class accounts for the background. This results in frame-wise predictions $\hat{A}_t \in \mathbb{R}^{W \times (C+1)}$.

To ensure consistency with the predictions of the offline teacher, we supervise the predicted CASS using the corresponding CAS segment $A_t^{\text{teacher}} \in \mathbb{R}^{W \times (C+1)}$ via an L_2 loss, $\mathcal{L}_{\text{cass}} = \frac{1}{W} \sum_{i=1}^{W} \left\| \hat{A}_t[i] - A_t^{\text{teacher}}[i] \right\|_2^2$. This loss encourages the online backbone to learn temporally precise, class-discriminative representations, facilitating more accurate online action localization.

3.4 Action Anticipatory Distillation

Window-level action anticipatory distillation strengthens the online student's temporal modeling by predicting which actions are likely to appear in the next W' frames using cues distilled from the offline teacher's CAS. This signal guides the model toward imminent action boundaries, enabling timely localization despite having no access to future frames.

Given the intermediate window feature $F_t \in \mathbb{R}^{W \times D}$, the anticipation head begins by reducing the feature dimension to $\frac{D}{4}$ through a linear transformation. The reduced features are then flattened and passed through two fully connected layers: the first uses a ReLU activation, while the second uses a sigmoid activation to produce a class-wise probability vector $\hat{y} \in [0,1]^{C+1}$, where C is the number of action classes and the additional class accounts for background.

During training, supervision is provided by the corresponding future segment of the CAS generated by the offline teacher. A threshold of 0.5 is applied to binarize this CAS segment, resulting in a multi-hot target vector $y \in \{0,1\}^{C+1}$ that indicates the set of actions expected to occur within the future window. The anticipation head is optimized using the binary cross-entropy loss, $\mathcal{L}_{\text{ant}} = -\sum_{c=1}^{C} \left[y_c \log(\hat{y}_c) + (1 - y_c) \log(1 - \hat{y}_c) \right]$, where y_c and $\hat{y}_c$ denote the teacher-derived target and predicted probability for action class c, respectively. This loss formulation enables the online model to benefit from future knowledge distilled from the offline model, improving its ability to reason about upcoming actions in an online streaming setup.

3.5 Actionness-Calibrated Anchor Decoding

Following prior works [11, 31], we adopt an anchor-based approach for action localization within our online model. At each sliding window, we define a fixed set of M temporal anchors with predefined scales. The anchor decoder is built upon a standard transformer decoder [37] that learns M anchor queries $\mathcal{Q} \in \mathbb{R}^{M \times D}$, and uses cross-attention to aggregate relevant information from the window feature $F_t \in \mathbb{R}^{W \times D}$, where W is the window length and D is the feature dimension.

The output of the transformer decoder is a set of anchor features $Z \in \mathbb{R}^{M \times D}$ representing encoded action proposals across the window.

To enhance the quality of attention in the anchor decoding process, we introduce a novel actionness-calibrated attention mechanism, which biases the attention weights using class-agnostic actionness scores derived from the Class Activation Sub-Sequence (CASS) predictor. A simple illustration of this method is presented in Figure 4 and discussed below.

Let the CASS prediction for the current window be denoted as $A_t \in \mathbb{R}^{W \times (C+1)}$, where the last channel $A_t^{(C+1)} \in \mathbb{R}^W$ corresponds to the background class. The actionness score sequence $r \in \mathbb{R}^{W \times 1}$ is computed as, $r = 1 - \sigma\left(A_t^{(C+1)}\right)$, where $\sigma(\cdot)$ is the sigmoid function. This formulation inversely maps low background probabilities to high actionness and vice versa.

To ensure that the calibration value is positive for high-actionness frames and negative for low-actionness frames, thus enhancing or suppressing attention accordingly, we transform the original scores using the following equation, $\bar{r} = r + \log(r)$. The resulting calibrated actionness signal $\bar{r} \in \mathbb{R}^{W \times 1}$ emphasizes frames with a higher likelihood of containing actions.

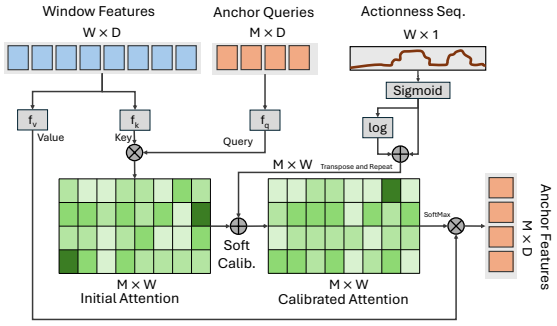

Fig. 4: Actionness sequence-based attention calibration for the anchor decoder. This module introduces an intentional bias, guiding the anchor features to emphasize high-quality action information from the most relevant frames while reducing influence from less informative ones.

This signal is incorporated into the transformer decoder's cross-attention mechanism by adding $\bar{r}$ as a bias term to the attention logits. Specifically, if $\text{Attn}(\mathcal{Q}, \mathcal{K}, \mathcal{V})$ denotes the standard scaled dot-product attention between queries $\mathcal{Q}$, keys $\mathcal{K}$, and values $\mathcal{V}$, we modify it as, $\text{Attn}_{\text{calibrated}}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \text{Softmax}\left(\frac{\mathcal{Q}\mathcal{K}^\top}{\sqrt{D}} + \bar{r}^\top\right)\mathcal{V}$, where $\mathcal{K}, \mathcal{V} \in \mathbb{R}^{W \times D}$ are derived from the window features F_t. This actionness-calibrated attention mechanism emphasizes frames with strong action likelihood during anchor decoding.

3.6 Auxiliary Anchor-Level Point Supervision

Our distillation framework transfers information from the offline teacher model to the online student across multiple levels. While this supervision effectively guides the student, the pseudo ground truth generated by the teacher is inherently imperfect, allowing errors to propagate. To make the framework more robust, we directly incorporate the actual point-level ground truth into the online model's training, ensuring stability against teacher-induced noise.

We introduce an Anchor-Level Point Prediction module that leverages point annotations as a weak yet reliable supervisory signal. The module contains a classification head to determine whether an action point lies within an anchor and its class, and a regression head to estimate the normalized distance between the anchor center and the point location. Both heads use a simple two-layer neural network with a ReLU activation in between. Based on prior evidence [24] and our pilot study (see supplementary material), point annotations collected in naturalistic settings tend to follow a Gaussian-like distribution centered around the action's midpoint. Leveraging this prior, the model is encouraged to reward anchors centered around action points and penalize those farther away.

The total point prediction loss is defined as $\mathcal{L}_{\text{pnt}} = \mathcal{L}_{\text{pc}} + \mathcal{L}_{\text{pr}}$, where $\mathcal{L}_{\text{pc}}$ is the cross-entropy classification loss, and the corresponding regression loss is formulated as $\mathcal{L}_{\text{pr}} = \frac{1}{N_p}\sum_{j=1}^{N_p}\left|\hat{d}_j - \frac{|c_a - p_j|}{l_a}\right|$, where $\hat{d}_j$ denotes the predicted normalized distance, c_a is the anchor center, p_j is the annotated point, and l_a is the anchor length. This auxiliary point supervision complements the teacher's guidance, yielding a more stable and noise-tolerant training process.

3.7 Anchor-Level Instance Supervision

In our distillation framework, the pseudo ground truth serves as direct supervision for the anchor-level instance prediction module, which is responsible for generating action proposals. Following the anchor-based approach of [11], this module employs two parallel heads: a classification head that determines whether an action occurs within a specific anchor and predicts its class, and a regression head that estimates temporal offsets and scaling factors to refine the anchor boundaries. The overall instance-level loss is formulated as $\mathcal{L}_{\text{ins}} = \mathcal{L}_{\text{ic}} + \mathcal{L}_{\text{ir}}$, where $\mathcal{L}_{\text{ic}}$ and $\mathcal{L}_{\text{ir}}$ represent the classification and regression losses, respectively. These objectives jointly encourage the model to identify the most relevant anchors and precisely localize their temporal extents. Throughout training, pseudo ground-truth labels produced by the offline teacher are used to supervise this instance prediction process.

3.8 Training and Inference Strategies

The total training objective is defined as a weighted combination of four loss components, $\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{ins}} + \beta\mathcal{L}_{\text{cass}} + \gamma\mathcal{L}_{\text{ant}} + \delta\mathcal{L}_{\text{pnt}}$. Here, $\mathcal{L}_{\text{ins}}$ is the anchor-level instance prediction loss, $\mathcal{L}_{\text{cass}}$ is the class-activation sub-sequence loss, $\mathcal{L}_{\text{ant}}$ is the anticipation loss, and $\mathcal{L}_{\text{pnt}}$ represents the point prediction loss. Definitions of these terms are provided in earlier sections. Empirically, we observe that setting $\alpha = \beta = \delta = 1$ yields the best results, and recommend tuning only γ, which governs the anticipation loss. This choice significantly reduces the hyperparameter tuning overhead.

During inference, following [11], our framework generates action proposals solely from anchor-level instance predictions, denoted as $\{(\hat{s}_k, \hat{e}_k, \hat{a}_k, \hat{p}_k)\}_{k=1}^{\hat{K}}$. To reduce redundant outputs in the online setting, we apply Online Non-Maximum

Suppression (ONMS) [35]. At each timestamp, ONMS discards proposals that have high temporal overlap with previously selected ones. Moreover, proposals with predicted end times beyond the current timestamp ($t < \hat{e}$) are also removed to ensure future, potentially more reliable predictions are not ignored.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on five benchmarks using only point-level supervision for training. (1) **THUMOS'14** [6]: 413 untrimmed sports videos with 20 actions, averaging ∼15 instances per video and large variation in action/video duration. (2) **EGTEA** [15]: 28 hours of egocentric kitchen videos (86 sessions, 32 subjects) annotated with 22 fine-grained actions, averaging ∼180 instances and 15.6 categories per video. (3) **HOI4D-Office** [23, 30]: 553 egocentric videos covering 12 office actions, with ∼15.5 instances per video, providing a structured evaluation for egocentric action understanding. (4) **FineAction** [22] and (5) **EPIC-Kitchens-100** [3]: two additional large-scale benchmarks covering more diverse and challenging action scenarios. For THUMOS'14, we use point annotations from [13] and generate point annotations for all other datasets using the same procedure.

Implementation Details. We primarily report standard mAP at multiple tIoU thresholds [11]; instance-level F1 [8] is additionally reported in the supplement. For features, we follow dataset protocols: THUMOS uses two-stream TSN features (Kinetics-pretrained) as in [11]; EGTEA uses Kinetics-pretrained I3D at 24 FPS with stride 12 [31]; and HOI4D-Office Tools uses pre-extracted CLIP-ViT features [30]. We adopt the offline backbone and post-processing of [49] for THUMOS and [4] for EGTEA/HOI4D-O. Our online model uses a D=1024 transformer encoder (3 blocks, 8 heads) and an anchor decoder (5 blocks, 4 heads). For sliding-window and anticipation, we use W=64, M=6, $\{4, 8, 16, 32, 48, 64\}$, W'=16 for THUMOS; W=24, M=7, $\{2, 4, 6, 8, 12, 16, 24\}$, W'=12 for EGTEA; and W=56, M=6, $\{4, 8, 16, 24, 48, 56\}$, W'=16 for HOI4D-O. We set γ=1.0 for THUMOS and 0.8 otherwise. For training, we use Adam (learning rate 1×10^{-4}, weight decay 1×10^{-4}). Additional details are in the supplement.

Baselines. As the first work to study POTAL, we establish both distillation-based and distillation-free baselines. For distillation-based baselines, we first use strong offline point-supervised TAL models to generate pseudo segments and then train state-of-the-art OnTAL models using these pseudo labels. We explored multiple offline teacher models (Table 5) and found HR-Pro [49] effective for sparse-action scenarios such as THUMOS'14, while TSASPC [4] performs better for dense or procedural settings like EGTEA and HOI4D-O. As online backbones, we evaluate OAT [11], MATR [35], and OAT-ONMS (OAT augmented with MATR's NMS to improve proposal quality). We additionally include a distillation-free baseline trained primarily with anchor-level point supervision (details in Supp. Sec. A).

Table 1: Comparison of model performance on THUMOS'14 dataset. We include results for offline fully-supervised, offline point-supervised, and online fully-supervised methods for reference. (* indicates our produced baselines.)

Method	mAP@tIoU (%)							AVG		
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	(0.1:0.5)	(0.3:0.7)	(0.1:0.7)
Offline + Full										
TCANet [28]	-	-	60.6	53.2	44.6	36.8	26.7	-	44.3	-
AFSD [16]	-	-	67.3	62.4	55.5	43.7	31.1	-	52.0	-
React [33]	-	-	69.2	65.0	57.1	47.8	35.6	-	55.0	-
ASL [32]	-	-	83.1	79.0	71.7	59.7	45.8	-	67.9	-
Offline + Point										
LACP [13]	75.7	71.4	64.6	56.5	45.3	34.4	21.8	62.7	44.5	52.8
SMBD [21]	-	-	66.0	57.9	47.0	36.0	22.0	64.2	45.7	-
RCTSI [45]	82.3	77.6	70.1	60.0	49.4	37.6	24.5	67.9	48.3	57.4
TSASPC [4]	70.2	68.8	67.1	63.2	58.0	51.6	43.1	65.5	56.6	60.4
HR-Pro [49]	85.6	81.6	74.3	64.3	52.2	39.8	24.8	71.6	51.1	60.3
QROT [20]	-	-	73.1	64.4	54.3	41.3	27.4	72.3	52.1	-
Online + Full										
CAG-QIL [9]	-	-	44.7	37.6	29.8	21.9	14.5	-	29.7	-
2PESNET [12]	-	-	47.4	39.8	31.4	21.8	14.0	-	30.9	-
SimOn [36]	-	-	57.0	47.5	37.3	26.6	16.0	-	36.9	-
OAT-OSN [11]	-	-	63.0	56.7	47.1	36.3	20.0	-	44.6	-
HAT [31]	-	-	62.0	57.0	48.0	36.5	20.7	-	44.8	-
MATR-ONMS [35]	-	-	70.3	62.7	52.1	38.6	23.7	-	49.5	-
OAT-ONMS [11, 35]*	-	-	66.1	61.3	52.4	41.6	26.0	-	49.5	-
Online + Point										
Distillation-Free Baseline*	59.9	45.8	31.8	18.4	10.3	5.6	2.0	33.3	13.6	24.8
HR-Pro + OAT-OSN *	59.1	55.8	49.3	41.5	29.7	16.3	7.1	47.1	28.8	37.0
HR-Pro + HAT*	58.4	55.2	48.9	39.3	28.2	16.5	6.6	46.0	27.9	36.2
HR-Pro + MATR-ONMS*	66.8	62.9	56.5	48.2	36.7	25.5	14.6	54.2	36.3	44.5
HR-Pro + OAT-ONMS*	66.6	63.7	58.7	49.3	40.1	28.7	16.6	55.7	38.7	46.3
OnPoint (Ours)	**73.6**	**70.3**	**63.9**	**56.3**	**45.2**	**30.8**	**17.5**	**61.9**	**42.7**	**51.1**

Table 2: Comparisons of performance on EGTEA and HOI4D-O datasets. (* indicates our produced baselines.)

Method	EGTEA						HOI4D-O					
	mAP@tIoU (%)						mAP@tIoU (%)					
	0.1	0.2	0.3	0.4	0.5	AVG[0.1:0.5]	0.1	0.2	0.3	0.4	0.5	AVG[0.1:0.5]
Offline + Point												
HR-Pro [49]	18.5	15.3	12.3	9.3	6.9	12.5	69.4	64.6	57.4	49.7	41.3	56.5
TSASPC [4]	48.6	46.0	41.5	36.4	30.6	40.6	77.7	73.5	68.0	59.8	51.2	66.0
Online + Full												
OAT-OSN [11]	24.2	22.7	20.5	16.3	10.9	18.9	55.5	53.2	48.9	42.0	33.8	46.7
HAT [31]	27.3	25.3	21.6	16.9	12.8	20.8	59.3	55.9	49.0	42.1	32.9	47.8
MATR-ONMS [35]	21.5	19.5	15.9	12.6	8.6	15.6	57.9	54.5	48.8	42.9	34.6	47.7
OAT-ONMS [11, 35]*	30.1	28.5	25.4	20.0	14.2	23.7	57.2	54.7	49.9	44.5	37.7	48.8
Online + Point												
Distillation-Free Baseline*	26.0	19.9	12.9	7.8	4.8	14.3	45.3	28.9	16.7	7.9	4.0	20.6
TSASPC + OAT-OSN*	24.4	21.5	17.5	12.0	7.1	16.5	55.6	49.8	42.0	33.3	24.2	41.0
TSASPC + HAT*	23.7	21.4	17.5	12.5	8.0	16.6	55.6	50.4	43.0	36.1	24.7	42.0
TSASPC + MATR*	20.6	18.4	15.1	11.1	7.1	14.5	55.8	51.5	44.8	32.9	22.7	41.6
TSASPC + OAT-ONMS*	27.0	24.4	20.8	16.0	10.3	19.7	55.9	52.6	45.2	35.2	22.6	42.3
OnPoint (Ours)	**30.9**	**28.6**	**24.5**	**18.9**	**12.6**	**23.1**	**58.2**	**53.5**	**47.4**	**37.4**	**26.4**	**44.6**

4.2 Evaluation on POTAL

We assess our approach on the POTAL task by comparing it with our baselines and existing fully-supervised, point-supervised, and online TAL methods.

THUMOS'14 and FineAction. As shown in Table 1, our method outperforms all baselines on the THUMOS'14 test set, achieving an average mAP gain of 4.8% across tIoU thresholds of 0.1–0.7, with improvements reaching up to 7.0% at individual thresholds. It also surpasses several early fully-supervised Online TAL methods and performs competitively with some fully-supervised

offline and point-supervised TAL approaches. On the more challenging FineAction benchmark (Supplementary Section B.1), featuring fine-grained, dense, and overlapping actions, OnPoint achieves 7.4% average mAP, outperforming the best baseline by 2.1%.

EGTEA, HOI4D-O, and EPIC-Kitchens-100. To validate generalizability across egocentric and procedural domains, we evaluate on EGTEA, HOI4D-O, and EPIC-Kitchens-100. As shown in Table 2, our method achieves consistent average mAP gains of 3.4% on EGTEA and 2.3% on HOI4D-O over tIoU thresholds of 0.1–0.5. Similar gains are observed on the larger EPIC-Kitchens-100 benchmark (Supplementary Section B.1), where the average mAP increases from 8.5% to 10.5%.

4.3 Model Analysis

Impact of Multi-Level Distillation. Our multi-level offline-to-online distillation framework includes frame-level class activation sub-sequence (CASS) alignment and window-level anticipation in addition to instance-level pseudo supervision. Removing these two components leads to a 4.4% average performance drop (Table 3). Ablating either individually also degrades performance, confirming the value of each. Notably, removing CASS also disables the actionness-calibrated attention module, which relies on CASS inputs.

Impact of Actionness Calibrated Attention. Removing the actionness sequence-based attention calibration (ASAC) from the anchor decoder reduces performance by 2.0% (Table 3). When removed from different combinations within our framework, performance consistently drops, highlighting its importance in refining anchor features through improved attention calibration. We further evaluate several variants of ASAC in Table 6. The main formulation described in Sec. 3.5 ($\bar{r} = r + \log(r)$) achieves the best performance, while ablation variants, including suppression-only ($\bar{r} = \log r$), enhancement-only ($\bar{r} = r$), simple scaling (multiplying r directly with the attention map), and direct suppression ($\bar{r} = -\sigma(A_t^{(C+1)})$), all lead to lower performance, highlighting the effectiveness of our proposed attention calibration formulation.

Table 3: Impact of proposed Window-Level Anticipation Distillation (WAD), Actionness Sequence-based Attention Calibration (ASAC), Class-Activation Sub-Sequence Distillation (CASD) on THUMOS'14 dataset.

Method	mAP@tIoU (%)					AVG
	0.1	0.2	0.3	0.4	0.5	[0.1:0.5]
Proposed	**73.6**	**70.3**	**63.9**	**56.3**	**45.2**	**61.9**
w/o WAD	71.1	67.9	62.3	54.6	44.3	60.0
w/o ASAC	72.6	69.2	63.1	53.6	41.3	59.9
w/o CASD & ASAC	71.1	67.7	61.7	52.3	40.0	58.5
w/o ASAC + WAD	69.2	65.3	60.0	52.8	42.5	58.0
w/o CASD & ASAC + WAD	69.2	65.1	60.2	52.1	40.8	57.5

Table 4: Impact of Anchor-level point supervision on THUMOS'14 dataset. APC = Anchor-Level Point Classification Head, APR = Anchor-Level Point Regression Head (* indicates that the corresponding loss is used as the primary loss (distillation-free setup) rather than an auxiliary loss)

Method	mAP@tIoU (%)					AVG
	0.1	0.2	0.3	0.4	0.5	[0.1:0.5]
Proposed	**73.6**	**70.3**	**63.9**	**56.3**	**45.2**	**61.9**
w/o APC	71.3	67.6	61.0	53.2	41.1	58.8
w/o APR	72.9	69.0	62.4	52.8	40.3	59.5
w/o APC + APR	70.1	67.3	61.1	52.5	40.3	58.4
only APC + APR*	59.9	45.8	31.8	18.4	10.3	33.3
only APC*	58.7	46.2	31.0	15.4	6.0	31.5

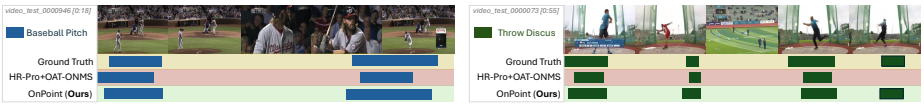

Fig. 6: Qualitative comparison of action localization results on two examples from the THUMOS'14 dataset: Baseball Pitch (left) and Throw Discus (right). We compare **OnPoint** and the baseline HR-Pro+OAT-ONMS and the ground truth annotations. For Baseball Pitch, both methods detect the action; however, our approach provides more complete and accurate temporal boundaries. For Throw Discus, the baseline fails to localize one subtle instance, whereas our method successfully detects all action instances with satisfactory boundary alignment.

Impact of Anchor-Level Point Prediction. Although auxiliary, the anchor-level point prediction improves training, with a 3.5% performance drop when removed (Table 4). Excluding either its classification or regression branch also leads to noticeable degradation, underscoring its contribution.

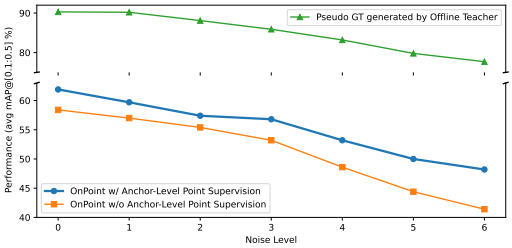

Fig. 5: Effect of offline-teacher noise on the online student. Anchor-level point supervision improves robustness to added increasing label noise and yields more stable performance than training without it by combining teacher guidance with point-level ground truth.

Moreover, this point-level supervision enhances the model's robustness against error propagation from the offline teacher. As illustrated in Figure 5, we introduced uniform noise at varying levels into the class activation sequence (CAS) outputs of the offline backbone, which in turn distorted the pseudo ground truth and propagated errors to the online student through distillation. Our model, equipped with anchor-level point supervision, maintained higher stability under increasing noise and consistently outperformed its variant without this component. This robustness stems from the model's dual reliance on both the teacher's guidance and the original point-level annotations, allowing it to learn more reliably even when teacher signals become noisy.

Framework Generalizability Across Offline Backbones. Table 5 illustrates the generalizability of our framework when applied to a range of existing offline backbone models. Although these backbones are not part of our main contribution, our framework treats them as modular and interchangeable components, enabling straightforward integration with future, more advanced offline

Table 5: Performance comparison of our framework using different offline backbone models, evaluated with and without our proposed components on THUMOS'14.

Method	mAP@tIoU (%)					AVG
	0.1	0.2	0.3	0.4	0.5	[0.1:0.5]
HR-Pro [49] (Whole)	66.6	63.7	58.7	49.3	40.1	55.7
+ **Our components**	**71.5**	**68.4**	**63.0**	**53.9**	**44.3**	**60.2**
HR-Pro [49] (RP)	69.8	66.2	60.3	51.6	41.7	57.9
+ **Our components**	**73.6**	**70.3**	**63.9**	**56.3**	**45.2**	**61.9**
SMBD [21]	68.6	63.4	56.8	48.3	35.9	54.6
+ **Our components**	**71.3**	**66.8**	**61.3**	**50.1**	**36.6**	**57.2**
LACP [13]	59.7	55.9	49.6	41.0	32.2	47.7
+ **Our components**	**64.0**	**60.3**	**54.4**	**44.4**	**32.9**	**51.2**
TSASPC [4]	61.5	59.9	55.5	48.8	38.7	52.9
+ **Our components**	**66.7**	**63.9**	**59.1**	**52.0**	**43.1**	**57.0**

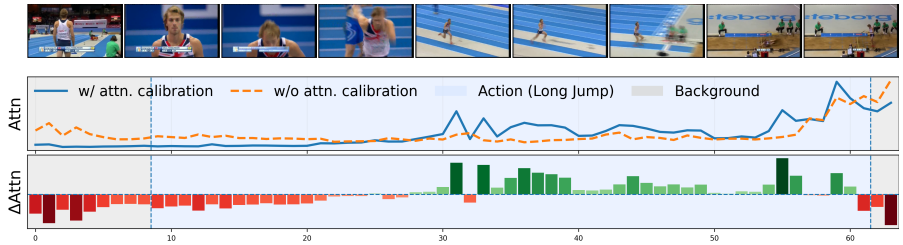

Fig. 8: Qualitative effect of attention calibration (last decoder layer attention example on THUMOS'14). Compared to the baseline, attention calibration sharpens attention over action-relevant past frames while suppressing background, yielding more discriminative temporal context and improved downstream performance (see Supp. Sec. B.9 for additional analysis).

architectures. Across all cases, incorporating our proposed modules consistently yields superior performance compared to the same backbones combined with a simple distillation baseline. This demonstrates the robustness and adaptability of our framework, showing that it can reliably enhance diverse off-the-shelf offline models regardless of their specific architecture.

Additional Analysis. For the **anticipation distillation**, the window-length analysis in Fig. 7 shows that a 16-frame anticipation window yields the best performance on THUMOS. We further compare alternative distillation strategies in the supplementary material Sec. B.7, where this simple fixed-window approach performs best. Regarding **post-processing**, our analysis (Supp. Sec. B.2) shows that replacing ONMS [35] with OSN [11] substantially reduces average mAP@tIoU[0.1:0.5] from 61.9 to 51.6, although **temporal promptness** improves, with Average Early Detection Time [11] (AEDT)@tIoU[0.1:0.5] decreasing from −0.17 to −1.53. This highlights an inherent trade-off: ONMS achieves better localization accuracy with slightly reduced promptness, whereas OSN improves temporal responsiveness at the cost of localization performance. The **inference efficiency analysis** in Supp. Sec. B.3 further shows that our online model remains computationally efficient (93M parameters, 2.88 GFLOPs, 312 FPS), achieving a favorable balance between speed and model capacity compared to heavier alternatives such as HAT and MATR, while remaining competitive with OAT across all efficiency metrics. Additional analyses, including **hyperparameter choices**, **sensitivity studies**, **CASS loss comparison**, and **instance-level F1 evaluation**, are provided in Supp. Sec. B.

Table 6: Comparison of Actionness Sequence-based Attention Calibration (ASAC) variants on THUMOS.

Method	avg mAP (%) @tIoU [0.1:0.5]
w/ ASAC (ours)	**61.9**
w/o ASAC	59.9
w/ ASAC suppress only	61.5
w/ ASAC enhance only	61.3
w/ ASAC simple scaling	58.3
w/ ASAC direct suppress	59.8

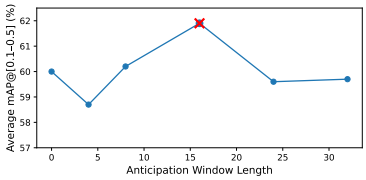

Fig. 7: Effect of anticipation window length on THUMOS14.

4.4 Qualitative Evaluation

Figure 6 compares OnPoint against the strongest baseline, HR-Pro+OAT-ONMS, on the THUMOS'14 test set. OnPoint provides more accurate localization of action instances. In the Baseball Pitch, both methods detect the action, but ours yields tighter and more complete temporal boundaries. In the Throw Discus case, the baseline misses a subtle instance, whereas our method correctly captures all occurrences with well-aligned boundaries. Figures 8 and 9 qualitatively illustrate the effect of our actionness-based attention calibration, which emphasizes key action frames while suppressing irrelevant ones, producing more discriminative anchor features and improving downstream performance.

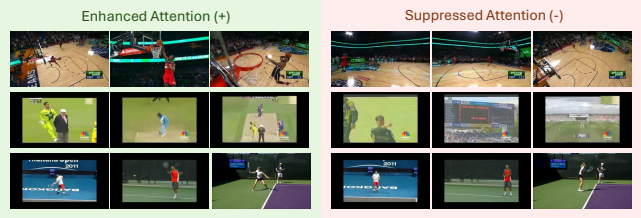

Fig. 9: OnPoint's actionness-based attention calibration enhances attention in frames with key action information while suppressing attention in irrelevant segments. Examples illustrate improved attention on action-relevant regions and reduced attention in non-informative intervals.

5 Conclusion

This paper introduces **Point-Supervised Online Temporal Action Localization (POTAL)**, a novel and practical problem setting for action localization in streaming videos trained with only point-level annotations. We propose **OnPoint**, which transfers full-video knowledge from an offline point-supervised teacher to a strictly online student via multi-level distillation, and further improves robustness with an actionness-guided anchor decoder and anchor-level point supervision. Experiments on five benchmarks show consistent gains over strong baselines, establishing OnPoint as a competitive and label-efficient solution for online temporal action localization and a solid starting point for future POTAL research.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant No. FW-HTF-2128743 and by the Office of Naval Research (ONR) under Grant No. N00014-21-1-2431. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or ONR.

References

1. Celdrán, F.J., Jiménez-Ruescas, J., Lobato, C., Salazar, L., Sánchez-Margallo, J.A., Sánchez-Margallo, F.M., González, P.: Use of augmented reality for training assistance in laparoscopic surgery: scoping literature review. Journal of Medical Internet Research **27**, e58108 (2025)
2. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
3. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision **130**(1), 33–55 (2022)
4. Du, D., Li, E., Si, L., Xu, F., Sun, F.: Timestamp-supervised action segmentation from the perspective of clustering. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. pp. 690–698 (2023)
5. Gupta, A., Mittal, G., Magooda, A., Yu, Y., Taylor, G.W., Chen, M.: Losa: long-short-range adapter for scaling end-to-end temporal action localization. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2092–2102. IEEE (2025)
6. Idrees, H., Zamir, A.R., Jiang, Y.G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The thumos challenge on action recognition for videos "in the wild". Computer Vision and Image Understanding **155**, 1–23 (2017)
7. Jiang, C., Dehghan, M., Jagersand, M.: Understanding contexts inside robot and human manipulation tasks through vision-language model and ontology system in video streams. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8366–8372. IEEE (2020)
8. Kang, H., Hyun, J., An, J., Yu, Y., Kim, S.J.: Actionswitch: Class-agnostic detection of simultaneous actions in streaming videos. In: European Conference on Computer Vision. pp. 383–400. Springer (2024)
9. Kang, H., Kim, K., Ko, Y., Kim, S.J.: Cag-qil: Context-aware actionness grouping via q imitation learning for online temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13729–13738 (2021)
10. Kim, H., Lee, S., Kang, H., Im, S.: Offline-to-online knowledge distillation for video instance segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 159–168 (2024)
11. Kim, Y.H., Kang, H., Kim, S.J.: A sliding window scheme for online temporal action localization. In: European Conference on Computer Vision. pp. 653–669. Springer (2022)
12. Kim, Y.H., Nam, S., Kim, S.J.: 2pesnet: Towards online processing of temporal action localization. Pattern Recognition **131**, 108871 (2022)
13. Lee, P., Byun, H.: Learning action completeness from points for weakly-supervised temporal action localization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13648–13657 (2021)
14. Li, J., Liu, X., Zhu, B., Jiao, J., Tomizuka, M., Tang, C., Zhan, W.: Guided online distillation: Promoting safe reinforcement learning by offline demonstration. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 7447–7454. IEEE (2024)

15. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: Joint learning of gaze and actions in first person video. In: Proceedings of the European conference on computer vision (ECCV). pp. 619–635 (2018)
16. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3320–3329 (2021)
17. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3889–3898 (2019)
18. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
19. Liu, H., Liu, Q., Wu, L., Shi, M., Cui, Z.: Offline-to-online: Case-based knowledge distillation with large language models for reinforcement learning. In: International Conference on Case-Based Reasoning. pp. 142–156. Springer (2025)
20. Liu, M., Wang, L., Zhou, S., Xia, K., Sun, X., Hua, G.: Boosting point-supervised temporal action localization through integrating query reformation and optimal transport. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 13865–13875 (2025)
21. Liu, M., Wang, L., Zhou, S., Xia, K., Wu, Q., Zhang, Q., Hua, G.: Stepwise multi-grained boundary detector for point-supervised temporal action localization. In: European Conference on Computer Vision. pp. 333–349. Springer (2024)
22. Liu, Y., Wang, L., Wang, Y., Ma, X., Qiao, Y.: Fineaction: A fine-grained video dataset for temporal action localization. IEEE transactions on image processing **31**, 6937–6950 (2022)
23. Liu, Y., Liu, Y., Jiang, C., Lyu, K., Wan, W., Shen, H., Liang, B., Fu, Z., Wang, H., Yi, L.: Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21013–21022 (2022)
24. Ma, F., Zhu, L., Yang, Y., Zha, S., Kundu, G., Feiszli, M., Shou, Z.: Sf-net: Single-frame supervision for temporal action localization. In: European conference on computer vision. pp. 420–437. Springer (2020)
25. Nie, M., Ding, D., Wang, C., Guo, Y., Han, J., Xu, H., Zhang, L.: Slowfocus: Enhancing fine-grained temporal understanding in video llm. Advances in Neural Information Processing Systems **37**, 81808–81835 (2024)
26. Patel, D., Babazaki, Y., Nagase, Y., Melvin, I., Min, M.R.: Distilling offline action detection models into real-time streaming models. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6205–6214 (2026)
27. Patsch, C., Wu, Y., Salihu, D., Zakour, M., Steinbach, E.: Tscl: Timestamp supervised contrastive learning for action segmentation. IEEE Robotics and Automation Letters (2024)
28. Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., Qiao, Y., Yan, J., Gao, C., Sang, N.: Temporal context aggregation network for temporal action proposal refinement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 485–494 (2021)
29. Reza, S., Song, X., Yu, H., Lin, Z., Moghaddam, M., Camps, O.: Reef: Relevance-aware and efficient llm adapter for video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2617–2628 (June 2025)

30. Reza, S., Sundareshan, B., Moghaddam, M., Camps, O.: Enhancing transformer backbone for egocentric video action segmentation. arXiv preprint arXiv:2305.11365 (2023)
31. Reza, S., Zhang, Y., Moghaddam, M., Camps, O.: Hat: History-augmented anchor transformer for online temporal action localization. In: European Conference on Computer Vision. pp. 205–222. Springer (2024)
32. Shao, J., Wang, X., Quan, R., Zheng, J., Yang, J., Yang, Y.: Action sensitivity learning for temporal action localization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13457–13469 (2023)
33. Shi, D., Zhong, Y., Cao, Q., Zhang, J., Ma, L., Li, J., Tao, D.: React: Temporal action detection with relational queries. In: European conference on computer vision. pp. 105–121. Springer (2022)
34. Shih, H.C.: A survey of content-aware video analysis for sports. IEEE Transactions on circuits and systems for video technology **28**(5), 1212–1231 (2017)
35. Song, Y., Kim, D., Cho, M., Kwak, S.: Online temporal action localization with memory-augmented transformer. In: European Conference on Computer Vision. pp. 74–91. Springer (2024)
36. Tang, T.N., Park, J., Kim, K., Sohn, K.: Simon: a simple framework for online temporal action localization. arXiv preprint arXiv:2211.04905 (2022)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
38. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. The Visual Computer **29**(10), 983–1009 (2013)
39. Wang, B., Zhao, Y., Yang, L., Long, T., Li, X.: Temporal action localization in the deep learning era: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(4), 2171–2190 (2023)
40. Wang, Q., Zhang, Y., Zheng, Y., Pan, P.: Rcl: Recurrent continuous localization for temporal action detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13566–13575 (2022)
41. Xia, H., Zhan, Y.: A survey on temporal action localization. IEEE Access **8**, 70477–70487 (2020)
42. Xia, K., Wang, L., Shen, Y., Zhou, S., Hua, G., Tang, W.: Exploring action centers for temporal action localization. IEEE Transactions on Multimedia **25**, 9425–9436 (2023)
43. Xia, K., Wang, L., Zhou, S., Hua, G., Tang, W.: Dual relation network for temporal action localization. Pattern Recognition **129**, 108725 (2022)
44. Xia, K., Wang, L., Zhou, S., Zheng, N., Tang, W.: Learning to refactor action and co-occurrence features for temporal action localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13884–13893 (2022)
45. Xia, Z., Cheng, J., Liu, S., Hu, Y., Wang, S., Zhang, Y., Dang, L.: Realigning confidence with temporal saliency information for point-level weakly-supervised temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18440–18450 (2024)
46. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: Proceedings of the IEEE international conference on computer vision. pp. 5783–5792 (2017)
47. Yoo, S., Reza, S., Tarashiyoun, H., Ajikumar, A., Moghaddam, M.: Ai-integrated ar as an intelligent companion for industrial workers: a systematic review. IEEE Access **12**, 191808–191827 (2024)

48. Zhang, C.L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: European Conference on Computer Vision. pp. 492–510. Springer (2022)
49. Zhang, H., Wang, X., Xu, X., Qing, Z., Gao, C., Sang, N.: Hr-pro: Point-supervised temporal action localization via hierarchical reliability propagation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38(7), pp. 7115–7123 (2024)
50. Zhang, Q., Fang, J., Yuan, R., Tang, X., Qi, Y., Zhang, K., Yuan, C.: Weakly supervised temporal action localization via dual-prior collaborative learning guided by multimodal large language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24139–24148 (2025)