WHERE TO LABEL AND HOW TO ADAPT AT TEST TIME FOR TEMPORAL ACTION SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies where to request frame-level annotations and how to use them to adapt a pretrained temporal action segmentation model at test time. To determine where to label, we propose an uncertainty-guided selection strategy that partitions each test video into consecutive regions with approximately equal uncertainty mass and queries one representative frame from each region. This design allocates annotations more densely to uncertain portions while retaining broad temporal coverage. To determine how to adapt, we use the queried labels as sparse semantic supervision and exploit their agreement with model predictions to dynamically regulate the adaptation trajectory across epochs. Experiments on standard benchmarks show consistent improvements under highly sparse annotation budgets. On Breakfast with MS-TCN, our approach improves accuracy from 65.4% to 81.9%, while querying only 0.16% of video frames on average.

1 INTRODUCTION

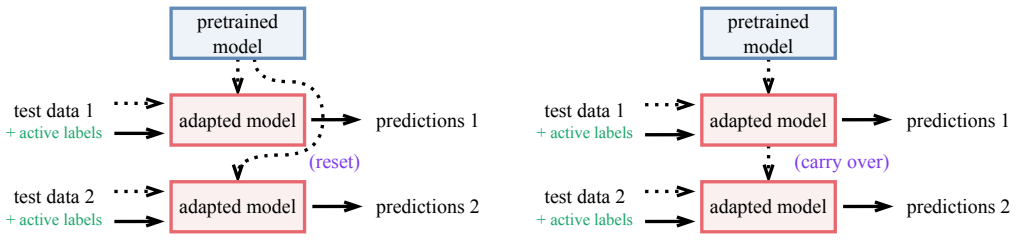
Temporal action segmentation (TAS) (Ding et al., 2023) has made considerable progress in parsing long, untrimmed videos into sequences of fine-grained actions. Recent temporal convolutional, attention-based, and generative models have substantially advanced this task (Farha & Gall, 2019; Yi et al., 2021; Liu et al., 2023; Singhania et al., 2023; Lu & Elhamifar, 2024). Nevertheless, these approaches typically learn a fixed model during training and apply it unchanged at inference, implicitly assuming that the knowledge acquired from the training data remains sufficient for the diverse conditions encountered at test time. SSTDA (Chen et al., 2020) challenged this assumption by showing that reducing discrepancies between training and testing TAS videos can significantly improve performance. Yet, its adaptation is performed during training with early access to target videos, limiting its applicability when target data are encountered only at inference.

Adapting a pretrained model directly to target data during inference is commonly referred to as test-time adaptation (TTA) (Liang et al., 2025). A variety of TTA methods have been proposed, relying on target normalization statistics, entropy minimization, consistency regularization, or pseudo-label self-training (Wang et al., 2021; Zhang et al., 2022; Niu et al., 2022). These adaptation signals are predominantly unsupervised and derived entirely from the model predictions, which may reinforce errors when the initial predictions are unreliable. As a remedy, active TTA (Gui et al., 2024; Lee et al., 2025) selectively queries labels for a limited number of test samples and uses the acquired supervision to guide model updates. This setting is practical when sparse feedback can be obtained from a human or a trusted stronger model, while densely labeling the test data is too costly.

This paper studies active TTA in an episodic setting, departing from existing formulations designed for an online setup (Gui et al., 2024; Lee et al., 2025). Fig. 1 illustrates the differences between two settings. Specifically, online adaptation carries model updates across successive test samples, assuming sufficient continuity for previously accumulated knowledge to remain beneficial. In TAS, however, consecutive test videos can exhibit distinct shifts in action composition and duration, making such cross-video adaptation unreliable[1]. This motivates us to adapt each test video independently and restore the source model before processing the next one. Two central questions regarding our active adaptation for TAS are: i) which frames should be labeled under a limited budget, and ii) how the acquired labels should be used to facilitate the adaptation.

[1]We empirically examine the effects of cross-video distribution shifts between TAS videos in Sec. 4.2, with further analyses provided in Appendix A.3.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Episodic adaptation (Ours) (b) Online adaptation

Figure 1: **Comparison of episodic and online active TTA.** Our episodic setting adapts each test video independently, and the model is reset before the next video. The online setting carries over adapted parameters across successive test videos. $\dashrightarrow$ indicates adaptation and $\rightarrow$ inference.

A common strategy for active selection is to query samples with the highest predictive uncertainty (Gal et al., 2017; Gui et al., 2024), and active selection has also been explored in TAS (Su & Elhamifar, 2024). In our post-deployment setting, high uncertainty frames are often temporally clustered around predicted action boundaries (see Fig. 2). Ranking frames by their absolute uncertainty values can cause multiple annotations to clutter within the same local region, leaving large portions of the video uncovered. To mitigate this, we introduce uncertainty-guided frame selection, which partitions each test video into consecutive regions with approximately equal uncertainty mass and selects one representative frame from each region. Annotations will be placed more densely where uncertainty is concentrated while preserving coverage elsewhere, balancing predictive uncertainty and temporal coverage under a fixed query budget.

The acquired labels can provide direct semantic supervision at the selected frames. Yet they only cover sparse locations in a long video, leaving most of the sequence without direct guidance. Inspired by (Li et al., 2021; Souri et al., 2022), we combine sparse classification supervision at the annotated frames with temporal smoothing over the full sequence, such that their influence can extend beyond the annotated locations. Other than providing semantic supervision, these annotated frames also offer a local signal of the adaptation process. At each adaptation epoch, agreement between the annotations and current predictions indicate that adaptation has reached a locally consistent solution, whereas disagreement suggests that errors remain unresolved. We leverage such signal to dynamically regulate changes across epochs, discouraging unnecessary updates where predictions already agree with the annotations while allowing greater flexibility where errors persist, thereby directing adaptation toward regions that remain unresolved.

In summary, we present a novel active adaptation framework for TAS. Its "active" component balances predictive uncertainty and temporal coverage; while its "adaptation" component combines sparse semantic supervision and dynamic trajectory regularization to guide model updates. The **main contributions** of our work are as follows: **1)** To the best of our knowledge, we provide the first formulation of active test-time adaptation for TAS, where a limited number of frame labels are queried to facilitate the adaptation. **2)** For active frame selection, we develop an uncertainty-guided strategy that allocates the query budget according to the temporal distribution of predictive uncertainty, concentrating annotations in uncertain portions while maintaining temporal coverage. **3)** For video-specific adaptation, we propose dynamic trajectory regularization that uses the queried labels not only as sparse semantic supervision but also as feedback to regulate updates across epochs. **4)** Extensive experiments on three TAS benchmarks and two model architectures show consistent improvements over strong competing comparison methods, achieving the best overall performance.

2 Related Work

Temporal Action Segmentation (TAS) has progressed from temporal convolutional architectures (Lea et al., 2017; Farha & Gall, 2019) to attention-based (Yi et al., 2021; Lu & Elhamifar, 2024) and generative models (Liu et al., 2023). These approaches improve temporal modeling and iterative prediction refinement but typically train a fixed model on source videos and apply it unchanged at inference. Other existing studies reduce training-time annotation requirements through timestamp (Li et al., 2021), weak (Richard et al., 2018; Kuehne et al., 2017; Ding & Yao, 2022b), semi-supervision (Ding & Yao, 2022a; Singhania et al., 2022) or unsupervised (Kumar et al., 2022;

Sener & Yao, 2018; Kukleva et al., 2019). More recent works have extended TAS towards more practical settings like online (Zhong et al., 2024), continual learning (Ding et al., 2024) and dataset condensation (Ding et al., 2025; Gauthier-Villar et al., 2026). Among these, Su & Elhamifar (2024) study active learning for TAS during training and rely on learned action prototypes to select representative samples for global model improvement. Yet such prototypes may not be present in a pretrained model to facilitate adaptation at test time, limiting the direct applicability of their selection strategy to our setting. In contrast, our method focuses on identifying frames that capture video-specific model characteristics and use their sparse labels for efficient adaptation.

Test-Time Adaptation (TTA) updates a trained model using target data available during inference. Existing methods commonly rely on unsupervised objectives such as entropy minimization (Wang et al., 2021; Niu et al., 2022; 2023; Seto et al., 2024) or pseudo-labels (Liang et al., 2020; Hou & Zheng, 2021). Active TTA instead queries a small number of target labels to provide stronger adaptation signals (Gui et al., 2024; Lee et al., 2025). However, existing active TTA methods primarily consider image-based settings, where test samples are treated independently and candidates are selected through uncertainty filtering (Gui et al., 2024) or Monte Carlo dropout (Lee et al., 2025). Our work introduces active TTA to untrimmed TAS videos, accounting for the temporal dimension by jointly considering the distribution of uncertainty over time and coverage across the full sequence.

3 METHOD

3.1 PROBLEM SETUP

Temporal Action Segmentation. Temporal action segmentation (TAS) (Ding et al., 2023) seeks to assign an action label to every frame of an untrimmed video. Given a video $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ of T frames, the TAS model f_θ produces frame-wise action labels $\hat{y}_t$:

$$\hat{y}_t = \arg\max_c \mathbf{p}_t(c), \quad \text{and} \quad \mathbf{p} = f_\theta(\mathbf{x}) \in \mathbb{R}^{T \times C}, \tag{1}$$

where $\mathbf{p}_t(c)$ denotes the probability of action c at frame t, and C is the number of action classes. Given the corresponding labels $\mathbf{y} = \{y_1, y_2, \ldots, y_T\}$, the TAS model (Farha & Gall, 2019; Yi et al., 2021) is commonly trained with a frame-wise classification loss $\mathcal{L}_{\text{cls}}$ along with temporal smoothing loss $\mathcal{L}_{\text{sm}}$ to discourage abrupt changes between neighboring predictions[2]. The two terms are balanced by a weighting coefficient λ:

$$\mathcal{L}_{\text{tas}} = \mathcal{L}_{\text{cls}}(x, y) + \lambda \mathcal{L}_{\text{sm}}(x). \tag{2}$$

Active Adaptation at Test Time. Given a pretrained TAS model f_θ, the task is to adapt it to a test video $\mathbf{x}$ under a query budget $K \ll T$. To account for variations in video length and action composition, we set $K = \eta N$, where N is the initial predicted number of segments and η a varying query ratio. The queried frame ratio is then $\rho = \eta N / T$. A selection policy returns an index set $\{s_1, s_2, \ldots, s_K\} \subset \{1, \ldots, T\}$, for which an annotation oracle provides the corresponding labels $\{y_{s_1}, y_{s_2}, \ldots, y_{s_K}\}$. These labels are used to adapt the model for E epochs, with the goal of improving its segmentation on the specific test video. We use superscript (e) to denote the e-th adaptation epoch, with $e = 0$ corresponding to the pretrained stage. Accordingly, $f_{\theta^{(E)}}$ is used to produce the final segmentation. The model is then reset to $f_{\theta^{(0)}}$ to adapt on the next video.

The task requires both 1) an effective policy for allocating the query budget across the video and 2) an adaptation strategy that fully exploits the acquired labels. We address the former through uncertainty-guided active frame selection (see Sec. 3.2) and the latter by adaptation with dynamic trajectory regularization (see Sec. 3.3), respectively.

3.2 UNCERTAINTY-GUIDED ACTIVE FRAME SELECTION

We posit that an effective active frame selection strategy for TAS should target uncertain predictions while maintaining temporal coverage across the video. Given the initial action probabilities $\mathbf{p}$, we quantify the predictive uncertainty at frame t using entropy:

$$H_t = -\sum_{c=1}^{C} \mathbf{p}_t(c) \log \mathbf{p}_t(c). \tag{3}$$

[2]Detailed loss formulations are provided in Appendix A.1.

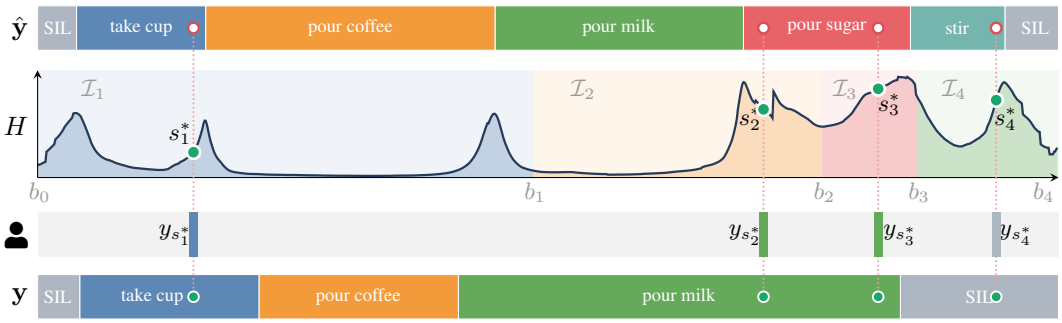


Figure 2: **Uncertainty Guided Active Frame Selection.** Given the initial predictions $\hat{\mathbf{y}}$ and frame-wise uncertainty H, we partition the video into $K = 4$ regions $\{\mathcal{I}_i\}_{i=1}^4$ with different colored background, each with approximately equal uncertainty mass. This partitioning produces narrower regions and denser queries where uncertainty is concentrated, and allocates broader regions and sparser queries to less uncertain portions. One representative frame s_i^* is selected within each region by minimizing its uncertainty-weighted temporal distance to all frames in that region, favoring locations near concentrations of uncertainty. The selected frames are queried from an oracle to obtain sparse labels $\{y_{s_i^*}\}_{i=1}^4$; the ground truth $\mathbf{y}$ is shown for reference.

Uncertainty-aware Temporal Coverage. A naive ranking of frames independently by H_t would concentrate queries within a small temporal region and provide limited coverage of the video. Hence, we distribute queries according to how uncertainty is distributed over time. With H_t along the temporal axis, we define the cumulative uncertainty mass up to frame t as:

$$A(t) = \sum_{\tau=1}^{t} H_\tau. \tag{4}$$

With a budget of K queries, we partition the sequence into consecutive regions containing approximately equal uncertainty mass. Let $b_0 = 0, b_K = T$, the intermediate boundaries are given by:

$$b_i = \min\{t \in 1, \ldots, T : A(t) \geq \frac{i}{K} A(T)\}, \qquad i \in 1, \ldots, K-1. \tag{5}$$

The resulting K regions are

$$\mathcal{I}_i = b_{i-1}+1, \ldots, b_i, \qquad i \in 1, \ldots, K. \tag{6}$$

Our partition naturally adapts the query density to the uncertainty distribution of the video. In highly uncertain portions, uncertainty accumulates rapidly, yielding more regions and thus more queries. Conversely, low-uncertainty portions span fewer, broader regions and receive fewer queries.

Uncertainty-weighted Frame Selection. The uncertainty mass partition specifies how the query budget should be distributed across the video, but not which frame should be annotated within each partition. To score a candidate frame $s \in \mathcal{I}_i$, we define an uncertainty-weighted proximity cost:

$$J_i(s) = \sum_{t \in \mathcal{I}_i} H_t \, d(t, s), \tag{7}$$

where $d(t, s)$ is a non-negative cost that increases with the temporal separation between frames t and s. The query frame is then selected according to:

$$s_i^* = \arg\min_{s \in \mathcal{I}_i} J_i(s). \tag{8}$$

Minimizing J_i favors candidates that are temporally close to highly uncertain frames, thereby biasing each query toward the more uncertain portions of the region rather than its confident portions. For simplicity, we implement the distance cost using the absolute temporal distance, *i.e.*, $d(t, s) = |t - s|$, and study other alternatives in Sec. 4.3.

After selecting the K representative frames, we query an annotation oracle for their ground-truth action labels $\{y_{s_i^*}\}_{i=1}^K$ to facilitate the model adaptation. An example of our active frame selection on a real test video from Breakfast (Kuehne et al., 2014) is shown in Fig. 2.

3.3 Adaptation with Dynamic Trajectory Regularization

As the obtained labels of the selected frames $\{y_{s_i^*}\}$ are sparse in time, we accordingly adapt the model with a sparsely supervised variant of the standard TAS objective. Specifically, the classification loss is imposed only at the annotated indices:

$$\mathcal{L}_{\text{spa}}(x, y_{s^*}) = -\frac{1}{K}\sum_{i=1}^{K} \log \mathbf{p}_{s_i^*}(y_{s_i^*}). \tag{9}$$

We retain temporal smoothing over the full sequence, yielding an adaptation loss as:

$$\mathcal{L}_{\text{adpt}} = \mathcal{L}_{\text{spa}}(x, y_{s^*}) + \lambda \mathcal{L}_{\text{sm}}(x). \tag{10}$$

The annotated labels provide direct semantic corrections at the selected frames, while full-sequence temporal smoothing encourages these corrections to extend to neighboring predictions.

Dynamic Trajectory Regularization. We further note that beyond providing semantic supervision, the actively acquired labels can also be used to assess how much further adaptation is needed through their agreement with the current predictions. The intuition is that if the model correctly classifies most annotated frames, large subsequent updates may be unnecessary. Conversely, persistent disagreement on these frames suggests a need for greater flexibility to update from the current state. A conceptual illustration of this process is provided in Fig. 3.

Starting from the pretrained prediction $\mathbf{p}^{(0)}$, let $\mathbf{p}^{(e)}$ denote the prediction obtained *before* backpropagation at the e-th adaptation epoch, *i.e.*, $\mathbf{p}^{(e)} = f_{\theta^{(e-1)}}(\mathbf{x})$. The sequence $\{\mathbf{p}^{(e)}\}_{e=1}^{E}$ defines the adaptation trajectory. Rather than using the pretrained prediction $\mathbf{p}^{(0)}$ as a fixed reference throughout adaptation, we use the preceding model state as a moving reference. Specifically, at each epoch e, we use the preceding prediction $\mathbf{p}^{(e-1)}$ as the moving reference and regularize $\mathbf{p}^{(e)}$ with:

$$\mathcal{L}_{\text{reg}}^{(e)} = \frac{1}{T}\sum_{t=1}^{T} w_t^{(e)} D_{\text{KL}}\left(\text{sg}\left[\mathbf{p}_t^{(e-1)}\right] \,\middle\|\, \mathbf{p}_t^{(e)}\right), \tag{11}$$

where $D_{\text{KL}}(\cdot||\cdot)$ denotes the KL divergence, $\text{sg}[\cdot]$ stops gradients through the preceding prediction, and $w_t^{(e)}$ controls the regularization strengths at frame t. To determine $w_t^{(e)}$, we first use the annotated frames to estimate the overall need for further adaptation. The accuracy of the preceding prediction state on these frames is:

$$\text{Acc}^{(e-1)} = \frac{1}{K}\sum_{i=1}^{K} \mathbb{1}\left[\hat{y}_{s_i^*}^{(e-1)} = y_{s_i^*}\right], \tag{12}$$

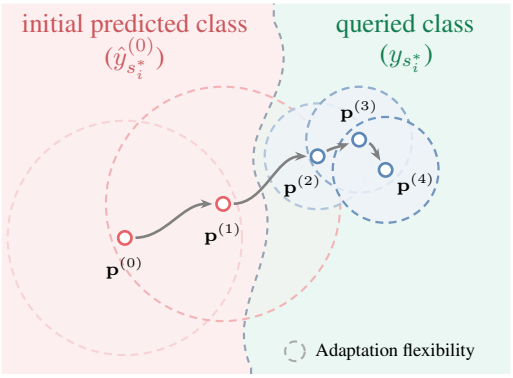


Figure 3: **Conceptual illustration of Dynamic Trajectory Regularization.** Dashed boundary separates the feature spaces between the initial predicted class and the queried ground-truth class. The sequence $\mathbf{p}^{(e)}$ traces the predictions of queried frames across adaptation epochs. When the prediction disagrees with the queried label, a weaker regularization is employed to permit larger updates (larger dashed circles). Once the prediction agrees with the queried label, stronger regularization (smaller dashed circles) constrains subsequent updates.

where $\mathbb{1}[\cdot]$ is the indicator function. A base regularization strength $\bar{w}^{(e)}$ is defined as:

$$\bar{w}^{(e)} = \alpha \begin{cases} 5, & \text{Acc}^{(e-1)} \geq \tau_{\text{high}}, \\ 1, & \tau_{\text{low}} < \text{Acc}^{(e-1)} < \tau_{\text{high}}, \\ 0.5, & \text{Acc}^{(e-1)} \leq \tau_{\text{low}}, \end{cases} \tag{13}$$

where α is a scaling coefficient and $\tau_{\text{low}}, \tau_{\text{high}}$ are lower and upper agreement thresholds, respectively. However, applying $\bar{w}^{(e)}$ uniformly ignores differences among the uncertainty mass regions $\mathcal{I}$ from Sec. 3.2. For each selected frame s_i^* in $\mathcal{I}_i$, we use its correctness under preceding prediction state to indicate whether predictions in that region should be preserved or allowed greater flexibility

Algorithm 1 Active Adaptation at Test Time for TAS

Input: Test video $\mathbf{x}$, pretrained model $f_{\theta^{(0)}}$, query budget K, adaptation epochs E
Output: Adapted model $f_{\theta^{(E)}}$

Feed forward $\mathbf{p}^{(0)} = f_{\theta^{(0)}}(\mathbf{x})$ ▷ Active Query (Sec. 3.2)
Compute uncertainty $\{H_t\}_{t=1}^{T}$ via Eq. (3) ▷ temporal coverage
Partition $\mathbf{x}$ into $\{\mathcal{I}_i\}_{i=1}^{K}$ via Eqs. (4) to (6)
for $i = 1, \ldots, K$ **do**
    Select s_i^* from $\mathcal{I}_i$ with Eqs. (7) and (8) ▷ active selection
    Query the oracle for $y_{s_i^*}$
end for
Initialize the moving reference with $\mathbf{p}^{(0)}$
for $e = 1, \ldots, E$ **do** ▷ Adaptation (Sec. 3.3)
    Feed forward $\mathbf{p}^{(e)}$ using model $f_{\theta^{(e-1)}}$
    Compute annotation agreement via Eq. (12)
    Determine $w_t^{(e)}$ via Eqs. (13) and (14)
    Update $f_{\theta^{(e-1)}}$ using Eq. (15)
end for
return $f_{\theta^{(E)}}$

to change. We therefore modulate $\bar{w}^{(e)}$ according to the correctness of annotated frames to obtain the frame-wise weight:

$$w_t^{(e)} = \begin{cases} 2\bar{w}^{(e)}, & t \in \mathcal{I}_i \text{ and } \hat{y}_{s_i^*}^{(e-1)} = y_{s_i^*}, \\ 0.5\bar{w}^{(e)}, & t \in \mathcal{I}_i \text{ and } \hat{y}_{s_i^*}^{(e-1)} \neq y_{s_i^*}. \end{cases} \tag{14}$$

A smaller weight value allows greater flexibility for the prediction to update, whereas a larger value more strongly constrains it to remain close to the preceding prediction. The final adaptation objective at epoch e combines sparse classification (Eq. (9)), full-sequence temporal smoothing (Eq. (17) in Appendix A.1), and the dynamic trajectory regularization (Eq. (11)):

$$\mathcal{L}_{\text{adpt}}^{(e)} = \mathcal{L}_{\text{spa}}(x, y_{s^*}) + \lambda \mathcal{L}_{\text{sm}}(x) + \mathcal{L}_{\text{reg}}^{(e)}(x, y_{s^*}, \hat{y}_{s^*}^{(e-1)}). \tag{15}$$

The entire pipeline of our active test-time adaptation procedure for TAS is provided in Algorithm 1.

4 Experiments

4.1 Dataset and Evaluation

Datasets. We conduct active test time adaptation on three widely used TAS benchmarks: Breakfast (Kuehne et al., 2014), 50Salads (Stein & McKenna, 2013), and GTEA (Fathi et al., 2011). Breakfast contains 1,712 videos of 10 breakfast preparation activities recorded across different environments and participants. 50Salads has 50 videos of salad preparation with 17 action classes. GTEA comprises 28 videos capturing daily kitchen activities performed by multiple subjects with 11 actions. We use the common I3D (Carreira & Zisserman, 2017) features as inputs.

Evaluation Metrics. We evaluate performance using the standard metrics of frame-wise accuracy (**Acc**), segmental edit score (**Edit**), and segmental **F1** scores at overlap thresholds of 10%, 25%, and 50%. Following the standard evaluation protocols, we use the official cross-validation splits for each dataset and report results averaged across all splits.

Baselines. As there are no existing TTA methods designed specifically for TAS, we construct the following baselines for comparison with the same query budget:

- ***Pretrained***: directly evaluates the source model on the target videos without test-time adaptation.
- ***Random***: randomly selects K frames for annotation, serving as a simple selection baseline.
- ***Equi*** (Rana & Rawat, 2023): selects K temporally equidistant frames for annotation, providing uniform coverage of the video.

Table 1: Comparison of active test-time adaptation methods on common TAS benchmarks.

	GTEA			50Salads			Breakfast		
Method	Acc	Edit	F1@10 / 25 / 50	Acc	Edit	F1@10 / 25 / 50	Acc	Edit	F1@10 / 25 / 50
MS-TCN (Farha & Gall, 2019)									
Pretrained	74.1	78.3	83.1 / 80.7 / 67.1	74.9	59.5	64.0 / 60.5 / 51.9	65.4	67.5	68.5 / 63.0 / 49.6
ρ	1.27%			0.11%			0.16%		
Random	77.2	79.6	85.8 / 83.2 / 70.0	77.2	59.4	66.9 / 63.1 / 54.0	75.8	70.8	74.2 / 68.5 / 53.0
Equi	77.5	80.4	84.9 / 82.3 / 69.8	79.7	64.0	71.1 / 68.8 / 59.1	78.2	74.2	77.4 / 71.5 / 56.1
Boundary	79.3	81.7	86.4 / 83.9 / 71.5	75.6	62.7	68.6 / 65.2 / 55.8	73.9	70.0	74.2 / 68.5 / 52.9
SimATTA	79.3	78.8	85.5 / 83.2 / 69.4	80.1	67.9	74.2 / 70.7 / 60.6	76.9	73.8	78.0 / 72.7 / 57.6
BiTTA	78.6	81.1	85.5 / 84.4 / 70.7	71.9	57.4	65.2 / 60.8 / 50.9	69.1	68.7	70.6 / 63.9 / 48.4
Ours	**79.5**	**83.2**	**88.0 / 85.5 / 73.4**	**82.7**	**69.3**	**77.4 / 74.1 / 64.9**	**81.9**	**79.5**	**81.8 / 77.8 / 65.5**
Full	82.7	83.8	89.0 / 86.9 / 76.2	86.2	74.4	81.0 / 79.0 / 71.7	91.7	88.5	90.7 / 89.3 / 84.3
ASFormer (Yi et al., 2021)									
Pretrained	76.5	82.5	87.7 / 84.9 / 74.5	76.7	69.6	74.9 / 71.0 / 61.0	72.8	75.1	76.2 / 71.0 / 58.9
ρ	1.41%			0.09%			0.14%		
Random	76.2	80.2	85.3 / 82.1 / 67.8	80.0	72.9	79.3 / 75.5 / 65.8	78.2	75.9	78.2 / 72.5 / 57.2
Equi	78.7	79.6	85.8 / 81.8 / 68.5	81.0	74.4	80.0 / 76.6 / 66.4	79.8	78.1	80.8 / 74.5 / 58.3
Boundary	78.9	83.2	89.1 / 86.0 / 73.2	79.7	74.9	80.6 / 77.0 / 66.5	78.0	76.6	80.4 / 75.1 / 59.4
SimATTA	79.7	85.0	90.9 / 88.6 / 74.0	80.7	**76.9**	82.0 / 78.5 / 68.5	78.2	78.4	81.8 / 76.2 / 61.0
BiTTA	73.1	76.6	83.0 / 79.4 / 65.9	77.0	70.2	76.4 / 71.9 / 61.4	73.6	74.4	76.3 / 70.5 / 55.7
Ours	**83.8**	**87.2**	**91.9 / 90.3 / 83.1**	**83.2**	76.3	**82.9 / 79.9 / 70.8**	**83.1**	**83.9**	**85.8 / 82.2 / 69.9**
Full	92.9	96.7	98.1 / 97.9 / 95.8	88.5	80.2	85.8 / 83.1 / 77.7	94.6	92.7	94.8 / 94.0 / 90.4

- *Boundary*: selects top-K frames based on the entropy from the action boundaries estimated from the pretrained model, serving as a boundary-centric selection baseline.
- *SimATTA* (Gui et al., 2024): follows its uncertainty-based selection strategy by applying k-means clustering to the 20% most uncertain frames and querying the K frames closest to each cluster center. Additional K frames with lowest uncertainty values are selected with pseudo-labels for adaptation.
- *BiTTA* (Lee et al., 2025): uses Monte Carlo dropout (Gal & Ghahramani, 2016) to estimate predictive variance across stochastic forward passes and queries K frames with the highest variance.
- *Full*: uses ground truth frame-wise label for adaptation, serving as the upper bound.

Implementation Details. We conduct experiments with two representative TAS models, MS-TCN (Farha & Gall, 2019) and ASFormer (Yi et al., 2021). With MS-TCN, we adopt a learning rate of $3e^{-4}$, $4e^{-5}$ and $7e^{-5}$ for Breakfast, GTEA and 50Salads, respectively. With ASFormer, we use $1e^{-4}$ for GTEA and 50Salads and $3e^{-4}$ for Breakfast. We employ a query ratio $\eta = 0.5$. We use $E = 5$ adaptation epochs for each test video, and reinitialize the model, optimizer and scheduler before adapting to the next one. More implementation details are provided in Appendix A.4.

4.2 Effectiveness

Main results and comparison. Table 1 reports results under the same annotation budget across three datasets and two backbone architectures. Relative to the pretrained models, our method improves consistently over all benchmarks with different segmentation models by a significant margin. The gains are especially significant on Breakfast, where Acc improves over the pretrained model by 16.5% with MS-TCN, and 10.3% with ASFormer. The increases in Edit and F1 scores further suggest that our adaptation improves temporal segment quality beyond the queried frames. Some visualizations and analyses can be found in Appendix A.4.

Interestingly, Equi (Rana & Rawat, 2023) stands out as a strong baseline, consistently improving over Random and some uncertainty-based counterparts. For example, on 50Salads with MS-TCN, it improves Edit and F1@50 over Random by 4.6% and 5.1%, respectively, highlighting the importance of broad temporal coverage for sparse annotation. Boundary performs less favorably despite targeting predicted transitions, where uncertainty is typically high, suggesting that selecting iso-

Table 2: Comparison between episodic and online adaptation on Breakfast. Episodic achieves better performance than online.

	Acc	Edit	F1@10 / 25 / 50
Pretrained	65.4	67.5	68.5 / 63.0 / 49.6
online	79.3	77.0	79.9 / 75.6 / 61.9
episodic	**81.9**	**79.5**	**81.8 / 77.8 / 65.5**

Table 3: Ablation study on Breakfast with ASFormer.

	Acc	Edit	F1@10 / 25 / 50
Pretrained	72.8	75.1	76.2 / 71.0 / 58.9
+ active labels	80.4	79.9	82.5 / 77.5 / 61.9
+ fixed KL	81.2	82.7	84.4 / 80.3 / 67.9
+ moving KL	82.4	83.4	85.3 / 81.2 / 68.6
+ dynamic KL	**83.1**	**83.9**	**85.8 / 82.2 / 69.9**

Table 4: Comparison of distance function choices for active frame selection on Breakfast.

	$d(t,s)$	Acc	Edit	F1@10 / 25 / 50
ℓ_1	$\lvert t-s\rvert$	**81.9**	**79.5**	81.8 / 77.8 / 65.5
ℓ_2	$(t-s)^2$	81.6	79.1	81.6 / 77.3 / 64.7
Exp	$\exp(\lvert t-s\rvert/20)-1$	81.8	79.3	**82.1** / 77.6 / 65.1
Log	$\log(1+\lvert t-s\rvert)$	81.1	79.4	**82.1 / 78.0 / 65.8**

lated uncertainty peaks may leave other informative regions uncovered. SimATTA (Gui et al., 2024) generally provides a competitive performance on all benchmarks. Its clustering of highly uncertain frames helps reduce redundant queries. However, restricting the candidate pool to a fixed proportion of the most uncertain frames does not explicitly account for the temporal distribution and may overlook less uncertain but temporally underrepresented regions. BiTTA (Lee et al., 2025) performs less consistently and this is likely because their variance-based selection through Monte Carlo dropout may be less reliable for temporally correlated frame predictions.

Episodic vs. online adaptation. We further evaluate our approach on the online setting where the parameters adapted on one test video are carried forward to initialize adaptation on the next. This differs from our episodic setting that resets the model to the pretrained parameters for each video, as illustrated in Fig. 1. In Table 2, we find our method remains effective in the online setting, and improves the pretrained model by 13.9% in Acc, 9.5% in Edit, and 11.4%, 12.6%, and 12.3% in F1. However, episodic adaptation consistently performs better, with further gains of 2.6% in Acc, 2.5% in Edit, and 1.9%, 2.2%, and 3.6% in F1. This differs from the observations of (Zhang et al., 2022) in active TTA for image classification where online adaptation outperforms episodic adaptation. We speculate this is because consecutive test videos in TAS can differ abruptly in action composition and action duration. Consequently, updates specialized to one video may not always provide benefit for the next video. Resetting the model avoids such cross-video accumulation. Additional comparisons and analyses of inter-video distribution shifts are provided in Appendix A.3.

4.3 Ablation and Analysis

Effect of active adaptation components. Table 3 progressively evaluates the components of our active adaptation framework. As we can see, introducing active labels greatly improves the pretrained model by a gain of 7.6% in Acc, showing the benefit of the sparse semantic supervision. Adding a fixed KL improves the segmental metrics, particularly Edit (2.8%) and F1@50 (6.0%). Replacing the fixed source reference with predictions from the preceding epoch consistently improves all metrics, showing the benefit of regularizing the evolving adaptation trajectory. Finally, dynamically adjusting the KL strength using feedback from queried labels yields further gains on all metrics, validating the effectiveness of our proposed dynamic trajectory regularization.

Effect of the annotation budget. Fig. 4 evaluates the sensitivity of our method to the query ratio η. Performance improves consistently as more annotations become available, with the largest marginal gain of about 4% when η increases from 0.25 to 0.5. Beyond that, performance continues to improve, although the gains are less pronounced. These results suggest that $\eta = 0.5$ provides a reasonable trade-off between annotation cost and segmentation performance, while demonstrating that the method remains effective across a broad range of budgets.

Effect of the temporal distance cost. The choice of $d(t,s)$ in Eq. (7) reflects different importance of frame uncertainty with respect to temporal distance, and we compare different options in Table 4. The ℓ_1 distance achieves the highest Acc and Edit scores of 81.9% and 79.5%, respectively,

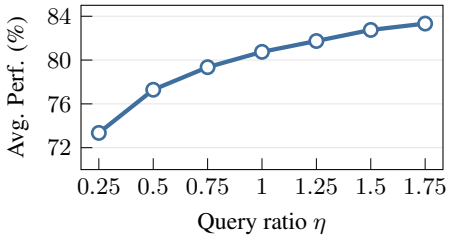


Figure 4: Influence of query ratio η on Breakfast. Performance generally grows with query ratio with $\eta = 0.5$ providing a nice trade-off.

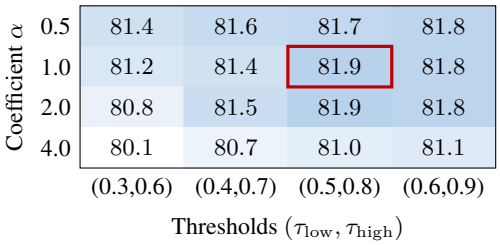


Figure 5: Averaged performance under varying coefficient α and thresholds $(\tau_{\text{low}}, \tau_{\text{high}})$ on GTEA. Default highlighted by the red box.

outperforming ℓ_2 by 0.3% and 0.4%. The logarithmic distance performs slightly better on the segmental metrics, achieving 78.0% F1@25 and 65.8% F1@50 compared with 77.8% and 65.5% for ℓ_1. We use ℓ_1 by default because it provides the strongest frame-wise accuracy and Edit score, while remaining competitive across F1 metrics and requiring no additional scaling parameter.

Effect of hyperparameters $\alpha, \tau_{\text{low}}, \tau_{\text{high}}$. We analyze the effect of the regularization scale α and agreement thresholds $(\tau_{\text{low}}, \tau_{\text{high}})$ on GTEA using MS-TCN in Fig. 5. For $\alpha < 2$ and threshold pairs ranging from $(0.4, 0.7)$ to $(0.6, 0.9)$, the performance varies only between 81.4 and 81.9. We choose $\alpha = 1, (\tau_{\text{low}}, \tau_{\text{high}}) = (0.5, 0.8)$ as the default configuration, which achieves 81.9. The same configuration is applied to all other datasets and backbones without dataset-specific tuning.

Are ground-truth boundaries beneficial? Table 5 examines whether access to precise action boundaries provides more useful supervision than our proposed frame-selection strategy. For each selected anchor frame, we locate its nearest ground-truth boundary and annotate the two immediately adjacent frames, thereby providing the action labels on both sides of the transition. Compared with the pretrained model, this boundary supervision yields substantial improvements, confirming that action boundaries provide useful adaptation signals.

Importantly, annotating precise action boundaries is more demanding than labeling individual frames. The annotator must examine the surrounding sequence, identify the actions before and after the transition, and locate the exact frame at which the action changes. This process resembles dense frame-wise annotation, which requires approximately 6× the annotation time of sparse frame labeling based on the estimate by Ma et al. (2020). Moreover, it is also pointed out by Ding & Yao (2022a) and Ding & Xu (2018), ground truth action transitions are often visually ambiguous, making these boundary adjacent frames difficult examples for the model to learn from during adaptation. Thus, they achieve only comparable performance to our frame selection strategy despite its substantially higher annotation cost.

Table 5: Comparison between annotating sampled frames and ground truth boundary (GTB). Ground truth provides comparable performance with ours yet requires 6× relative annotation time.

	Rel. Time	Acc	Edit	F1 @ 10 / 25 / 50
Pretrained	-	65.4	67.5	68.5 / 63.0 / 49.6
Ours	1×	**81.9**	79.5	81.8 / 77.8 / 65.5
GTB	6×	81.8	**80.7**	**82.8 / 78.7 / 67.4**

5 Conclusion

This paper introduces the first framework for active adaptation of temporal action segmentation models at test time, focusing on two key challenges: selecting query frames under a limited annotation budget and effectively exploiting the resulting sparse supervision. To this end, we propose an uncertainty-guided selection strategy that distributes annotations according to the temporal distribution of predictive uncertainty while maintaining broad temporal coverage, together with dynamic trajectory regularization that uses the queried labels both as semantic supervision and as feedback to regulate the adaptation trajectory across epochs. Our extensive experiments and analyses show consistent improvements in segmentation performance, highlighting the broader potential of selective supervision for adapting pretrained models at test time.

AI USE STATEMENT

We used generative AI tools to provide feedback on research methodology and experimental design, assist with code development and debugging, discuss the interpretation of experimental results, and support the preparation of scientific figures. These tools were also used to summarize relevant literature, identify related research directions, improve the structure and readability of the manuscript, and refine portions of the text.

Generative AI was not used to generate synthetic datasets, formulate or prove mathematical claims, translate research materials, or perform qualitative data analysis. All AI-assisted code was inspected and tested by the authors, and all suggested methods, experiments, interpretations, citations, figures, and written content were independently reviewed and verified. The authors take full responsibility for the methods, results, claims, and final content of this work.

ETHICS STATEMENT

All annotation queries in this work are simulated by revealing existing ground-truth labels from publicly available temporal action segmentation benchmarks. No human participants or annotators were recruited, no new personal data were collected, and no annotator behavior was studied. Accordingly, the reported experiments do not constitute a new human-subject study. The benchmark datasets are used in accordance with their established research protocols and licenses.

REPRODUCIBILITY STATEMENT

We provide the complete formulation of our active test-time adaptation framework, including uncertainty-guided frame selection and dynamic trajectory regularization, in Sec. 3. The datasets, evaluation protocols, baselines, and implementation settings are described in Sec. 4, with additional training and adaptation hyperparameters provided in Appendix A.4.

REFERENCES

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *CVPR*, 2020.

Guodong Ding and Angela Yao. Leveraging action affinity and continuity for semi-supervised temporal action segmentation. In *ECCV*, 2022a.

Guodong Ding and Angela Yao. Temporal action segmentation with high-level complex activity labels. *TMM*, 2022b.

Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE TPAMI*, 2023.

Guodong Ding, Hans Golong, and Angela Yao. Coherent temporal synthesis for incremental action segmentation. In *CVPR*, 2024.

Guodong Ding, Rongyu Chen, and Angela Yao. Condensing action segmentation datasets via generative network inversion. In *CVPR*, 2025.

Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, 2018.

Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, 2019.

Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, 2017.

Artheme Gauthier-Villar, Guodong Ding, and Angela Yao. Adaptive latent trajectory anchoring for action segmentation dataset condensation. In *ECCV*, 2026.

Shurui Gui, Xiner Li, and Shuiwang Ji. Active test-time adaptation: Theoretical analyses and an algorithm. In *ICLR*, 2024.

Yunzhong Hou and Liang Zheng. Visualizing adapted knowledge in domain transfer. In *CVPR*, 2021.

Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.

Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *CVIU*, 2017.

Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *CVPR*, 2019.

Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Unsupervised action segmentation by joint representation learning and online clustering. In *CVPR*, 2022.

Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017.

Taeckyung Lee, Sorn Chottananurak, Junsu Kim, Jinwoo Shin, Taesik Gong, and Sung-Ju Lee. Test-time adaptation with binary feedback. In *ICML*, 2025.

Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *CVPR*, 2021.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *IJCV*, 2025.

Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *ICCV*, 2023.

Zijia Lu and Ehsan Elhamifar. Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. In *CVPR*, 2024.

Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *ECCV*, 2020.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *ICLR*, 2023.

Aayush J Rana and Yogesh S Rawat. Hybrid active learning via deep clustering for video action detection. In *CVPR*, 2023.

Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *CVPR*, 2018.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *NeurIPS*, 2020.

Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *CVPR*, 2018.

Skyler Seto, Barry-John Theobald, Federico Danieli, Navdeep Jaitly, and Dan Busbridge. Realm: Robust entropy adaptive loss minimization for improved single-sample test-time adaptation. In *WACV*, 2024.

Dipika Singhania, Rahul Rahaman, and Angela Yao. Iterative contrast-classify for semi-supervised temporal action segmentation. In *AAAI*, 2022.

Dipika Singhania, Rahul Rahaman, and Angela Yao. C2f-tcn: A framework for semi-and fully-supervised temporal action segmentation. *IEEE TPAMI*, 2023.

Yaser Souri, Yazan Abu Farha, Emad Bahrami, Gianpiero Francesca, and Juergen Gall. Robust action segmentation from timestamp supervision. In *BMVC*, 2022.

Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *UbiComp*, 2013.

Yuhao Su and Ehsan Elhamifar. Two-stage active learning for efficient temporal action segmentation. In *ECCV*, 2024.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *BMVC*, 2021.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2022.

Qing Zhong, Guodong Ding, and Angela Yao. Onlinetas: An online baseline for temporal action segmentation. In *NeurIPS*, 2024.

A Appendix

A.1 TAS loss functions

Given a video $\{x_1, x_2, \ldots, x_T\}$, the ground truth labels $\{y_1, y_2, \ldots, y_T\}$, and the predicted action probability $\mathbf{p}_t(c)$ for action c at frame t, the classification loss is defined as:

$$\mathcal{L}_{\text{cls}}(x, y) = -\frac{1}{T}\sum_{t} \log \mathbf{p}_t(y_t), \tag{16}$$

while the smoothing loss (Farha & Gall, 2019) is written as:

$$\mathcal{L}_{\text{sm}}(x) = \frac{1}{TC}\sum_{t,c} \tilde{\Delta}^2_{t,c}, \quad \tilde{\Delta}_{t,c} = \begin{cases} \Delta_{t,c} : \Delta_{t,c} \leq \tau \\ \tau \quad \text{:otherwise} \end{cases} \quad \text{and} \quad \Delta_{t,c} = \left|\log \mathbf{p}_t(c) - \log \mathbf{p}_{t-1}(c)\right|, \tag{17}$$

and $\tau = 4$ as per (Farha & Gall, 2019) is used to truncate excessively large temporal differences.

A.2 Conventional TTA techniques not directly applicable

Conventional TTA methods are predominantly developed for image classification, where target data arrive as batches or streams of independent samples (Wang et al., 2021; Niu et al., 2022). TAS presents a different adaptation unit: each test sample is a variable-length video containing strongly correlated frame predictions, and the objective is to improve the segmentation of the complete sequence. Treating individual frames as independent target samples neglects their temporal dependencies, whereas treating a full video as a single sample provides no population of independent observations from which to estimate a target distribution.

Moreover, several widely used TTA methods adapt batch-normalization statistics or their affine parameters (Schneider et al., 2020; Wang et al., 2021). However, such a mechanism is not directly applicable to the TAS backbones considered in our experiments. This is because standard TAS models like MS-TCN (Farha & Gall, 2019) and ASFormer (Yi et al., 2021) do not provide the batch-normalization statistics or parameters required by such methods. Prediction-based TTA objectives could still be applied by updating other model parameters, but their supervision is derived entirely from the model's own predictions. In TAS, prediction errors are temporally correlated and may persist over extended portions of a video, creating a risk of reinforcing incorrect segments during adaptation. These differences motivate the use of sparse frame-level annotations and sequence-aware regularization in our setting.

Table 6: Comparison between online and episodic adaptation across GTEA and 50Salads.

	GTEA			50Salads		
Setting	Acc	Edit	F1 @ 10 / 25 / 50	Acc	Edit	F1 @ 10 / 25 / 50
Pretrained	74.1	78.3	83.1 / 80.7 / 67.1	74.9	59.5	64.0 / 60.5 / 51.9
online	78.3	80.2	84.4 / 82.5 / 73.6	79.8	65.0	70.7 / 67.2 / 58.7
episodic	**79.5**	**83.2**	**88.0 / 85.5 / 73.4**	**82.7**	**69.3**	**77.4 / 74.1 / 64.9**

A.3 Online vs. Episodic

Additional comparison on other two benchmarks between online and episodic settings are provided in Table 6. Online TTA continuously updates the model and carries the adapted parameters from one test sample to the next, which is most beneficial when neighboring samples exhibit some distributional continuity.

This assumption may not hold for TAS. As shown in Table 7, different Breakfast activity videos contain distinct action sets, despite sharing an overall label vocabulary. To quantify these differences, we compute the pairwise KL divergence between the frame-wise action distributions between activity pairs. For each activity, we obtain its action distribution by first computing the normalized frame-level action frequencies for each video and then averaging these distributions across all videos of that activity. Fig. 6 shows generally large divergences across activities, frequently exceeding 16, while lower values occur primarily between semantically related activities, such as "*friedegg*" and "*scrambledegg*". The asymmetry of the heatmap further indicates that the magnitude of the distribution shift depends on the direction of the activity transition.

Therefore, a model first adapted on a "*coffee*" video becomes specialized to its constituent actions, which differ considerably from those in a possible subsequent "*scrambledegg*" video. Consequently, the adaptation acquired from the "*coffee*" video may offer little benefit to the "*scrambledegg*" video and may even bias its predictions toward actions emphasized by the preceding activity.

A.4 Hyperparameters and More results

Hyperparameter. We provide the detailed hyperparameters used in our experiments in Table 8.

Effectiveness of dynamic KL. We further evaluate the generality of dynamic trajectory regularization by combining it with different frame-selection strategies in Table 9. Dynamic KL consistently improves Edit and all F1 scores across every strategy, with gains of 3.8%-7.5% in Edit and 5.7%-10.6% in F1@50. It also improves frame-wise accuracy for three of the four strategies, while pro-

Table 7: Action composition of the videos in Breakfast. Different activities involve different subsets of actions.

Activity	Action composition
cereals	pour cereals, pour milk, take bowl, stir cereals
coffee	pour coffee, pour milk, take cup, pour sugar
friedegg	add saltnpepper, crack egg, take plate, pour oil, butter pan, take eggs, put egg2plate, fry egg
juice	squeeze orange, pour juice, cut orange, take glass, take squeezer, take plate, take knife
milk	spoon powder, pour milk, stir milk, take cup
pancake	fry pancake, stir dough, crack egg, pour milk, spoon flour, take plate, butter pan, pour flour, pour oil, pour dough2pan, put pancake2plate, take bowl
salat	cut fruit, peel fruit, put fruit2bowl, take knife, take bowl, stir fruit, take plate
sandwich	smear butter, put toppingOnTop, cut bun, take topping, take knife, take plate, take butter, put bunTogether
scrambledegg	stirfry egg, put egg2plate, stir egg, crack egg, add saltnpepper, pour egg2pan, take plate, butter pan, pour oil, take bowl
tea	pour water, add teabag, take cup

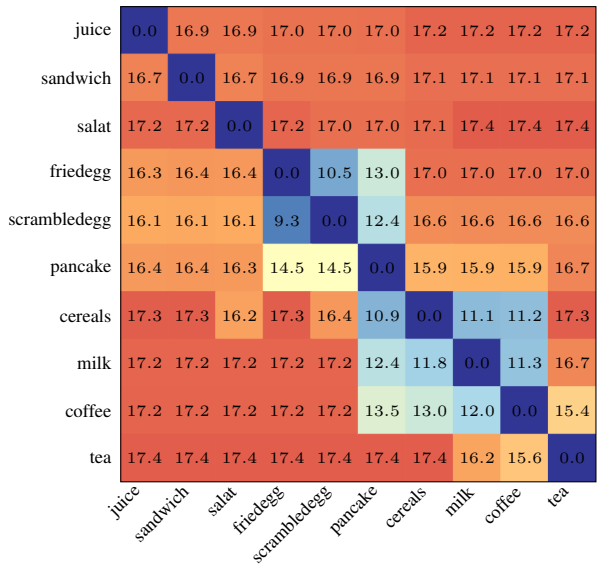


Figure 6: Pairwise KL divergence between the action distributions of Breakfast activities.

ducing a marginal decrease of 0.2% for SimATTA. These consistent segment-level improvements indicate that dynamic KL is broadly complementary to different selection strategies and effectively stabilizes adaptation under sparse supervision.

More results on GT boundaries. We provide the results for annotating the groundtruth boundaries on GTEA and 50Salads in Table 10. Similar to what we reported in the main paper, GTB achieves comparable performance to ours yet requires 6× the relative annotation time compared to ours.

Effect of adaptation epochs E. Fig. 7 studies the effect of varying the number of adaptation epochs. Performance improves steadily during the initial epochs, demonstrating that the queried annotations provide effective supervision for adapting the pretrained model. The improvement becomes marginal after 5 epochs, indicating that the model has largely converged under the sparse supervi-

Table 8: Parameters used for active adaptation in our experiments.

	MS-TCN			ASFormer			Remarks
	GTEA	50Salads	Breakfast	GTEA	50Salads	Breakfast	
lr	$4e^{-5}$	$7e^{-5}$	$3e^{-4}$	$1e^{-4}$	$1e^{-4}$	$3e^{-4}$	Learning rate
E	5			5			Adaptation epochs
η	0.5			0.5			Query ratio
λ	0.15			0.15			Smoothing loss weight
α	1			1			Dynamic KL coefficient
$(\tau_{\text{low}}, \tau_{\text{high}})$	(0.5, 0.8)			(0.5, 0.8)			Agreement thresholds

Table 9: Performance of the proposed dynamic trajectory regularization with different sampling strategies on Breakfast using the MS-TCN backbone.

Method	Acc	Edit	F1@ 10 / 25 / 50	Method	Acc	Edit	F1@ 10 / 25 / 50
Equi	78.2	74.2	77.4 / 71.5 / 56.1	Boundary	73.9	70.0	74.2 / 68.5 / 52.9
+ dynamic KL	**81.1**	**78.1**	**80.8 / 76.3 / 63.4**	+ dynamic KL	**78.1**	**77.5**	**80.6 / 76.4 / 63.5**
SimATTA	**76.9**	73.8	78.0 / 72.7 / 57.6	BiTTA	69.1	68.7	70.6 / 63.9 / 48.4
+ dynamic KL	76.7	**77.6**	**80.2 / 76.0 / 63.3**	+ dynamic KL	**72.9**	**74.6**	**76.8 / 71.6 / 58.5**

Table 10: Comparison between our proposed active selection and using ground truth boundary (GTB) annotation. For each selected anchor, we first locate its nearest ground truth boundary, and then annotate the two frames adjacent to the boundary.

Sampling	Rel. Time	GTEA			50Salads		
		Acc	Edit	F1 @ 10 / 25 / 50	Acc	Edit	F1 @ 10 / 25 / 50
Ours	1×	**79.5**	**83.2**	**88.0 / 85.5 / 73.4**	82.7	**69.3**	**77.4 / 74.1 / 64.9**
GTB	6×	77.9	79.5	85.4 / 82.1 / 71.0	**83.0**	68.6	75.8 / 72.9 / 64.3

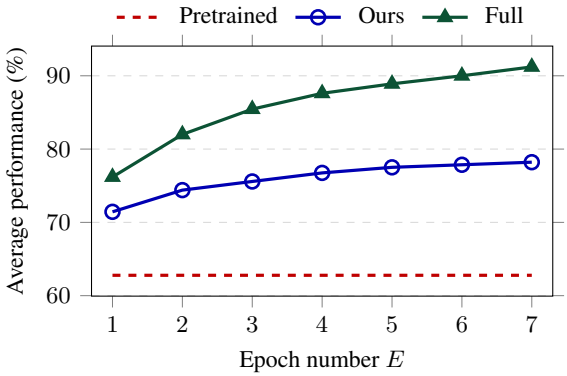


Figure 7: Effect of the number of adaptation epochs on average performance with Breakfast.

sion. We therefore use $E = 5$ by default, providing a favorable trade-off between performance and computational cost.

Visualizations. Fig. 8 qualitatively illustrates two example videos from Breakfast with our active adaptation framework. In Fig. 8(a), the initial prediction contains several spurious context-driven segments near the end of the video. Specifically, '*pour sugar*' and '*stir coffee*' are optional actions that commonly co-occur in coffee preparation but are absent from this particular video, suggesting that the model relies on activity-level contextual priors rather than the observed visual evidence. With only four queried frames, adaptation removes these erroneous segments, extends the '*pour milk*' prediction, and produces a segmentation that closely matches the ground truth. Fig. 8(b)

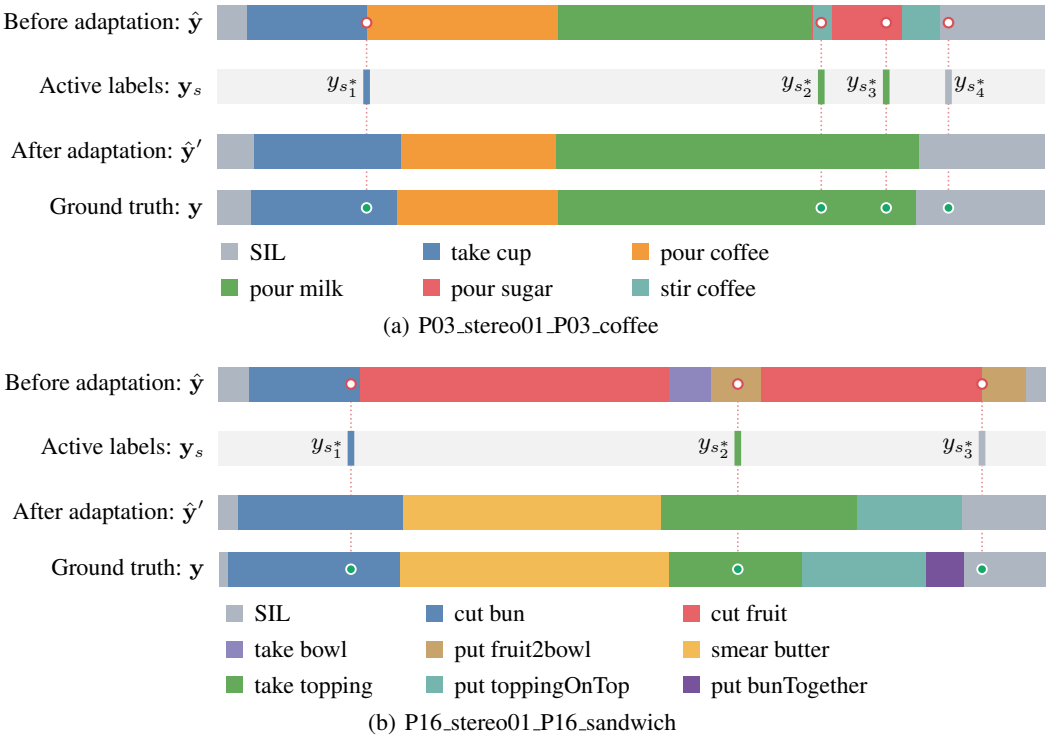


Figure 8: Visualization of active adaptation on Breakfast example videos. The initial prediction $\hat{\mathbf{y}}$ is adapted using sparse labels $\mathbf{y}_s$ at the selected frames, producing $\hat{\mathbf{y}}'$. The full ground truth $\mathbf{y}$ is shown for reference. a) shows active annotation can help remove spurious contextual correlation; while b) shows that correcting a small number of errors can help recover other unannotated actions.

shows a more substantial prediction error for a video depicting "*sandwich*" preparation. The initial segmentation is largely dominated by '*cut fruit*' and fails to recover the correct action sequence. After adaptation with only three annotated frames, the model recovers a more accurate temporal structure, including '*smear butter*', '*take topping*', and '*put toppingOnTop*'. Notably, '*smear butter*' and '*put toppingOnTop*' are recovered despite receiving no direct annotation. This suggests that sparse supervision can improve predictions beyond the queried frames, enabling the recovery of unannotated actions through learned temporal and semantic dependencies within the activity.

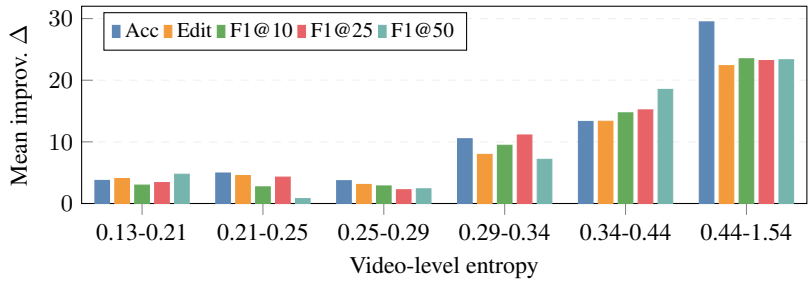


Figure 9: Performance improvement across video-level entropy bins on Breakfast. Bars report the mean improvement after adaptation. Higher-entropy videos generally exhibit larger performance gains across all metrics.

Is predictive entropy a meaningful signal for active TTA? For each video, we first compute the predictive entropy of every frame and then average these frame-level entropies to obtain a single video-level entropy score. We then group the test videos into six quantile bins according to this video-level score. As shown in Fig. 9, videos in the lower entropy bins exhibit relatively small

improvements across all metrics, while higher-entropy videos achieves more significant gains. Although small fluctuations are observed among the lower-entropy bins, the overall trend is consistent across metrics. This observation provides empirical support for using entropy as the guiding signal in our selection design.

A.5 Limitation and Outlooks

Our experiments simulate the annotation process using existing ground truth frame labels and do not account for the potential annotation noise, ambiguity, or variability in real human feedback. Future work could study active test-time adaptation with real annotators and develop adaptation strategies that remain robust to imperfect supervision. More broadly, extending this framework to richer forms of test-time feedback and other temporally structured video tasks presents an interesting direction for post-deployment model adaptation.