Boundary-Aware Progress-Balanced Prefix Learning for Strictly Online Temporal Action Segmentation

Anonymous authors
Paper under double-blind review

Abstract

Temporal action segmentation assigns an action label to every frame of an untrimmed video. Most segmentation models are trained with complete videos, whereas strictly online inference can access only a growing observed prefix. Directly applying a full-sequence model to prefix inference substantially degrades segment quality, and causal constraints recover only part of the loss. We characterize this mismatch as temporal context distribution shift and address it by reconstructing training prefixes and their supervision. Our method first balances prefix endpoints across observation progress, then increases sampling near action transitions, and finally weights frame-level supervision toward the prefix endpoint. The approach modifies training but introduces no additional inference branch. With FACT as the backbone, it reaches F1@0.5 scores of 68.42% on GTEA and 39.68% on 50Salads, improving over causal adaptation by 9.32 and 26.86 points. Applying the strategy to ASFormer also improves F1@0.5 over causal adaptation on both datasets, indicating that its benefits can extend to other attention-based backbones. These results show that aligning prefix coverage and supervision is important when adapting attention-based segmentation models to strictly online inputs.

1 Introduction

Temporal action segmentation (TAS) assigns an action label to every frame of an untrimmed video and recovers a sequence of contiguous action segments (Ding et al., 2024). Modern approaches model temporal structure with multi-stage convolutions, boundary refinement, diffusion, or frame–action attention (Farha & Gall, 2019; Ishikawa et al., 2021; Liu et al., 2023; Lu & Elhamifar, 2024). Most follow an offline protocol and can access the complete video during training and inference. This assumption prevents direct deployment when frames arrive sequentially.

Strictly online TAS requires a prediction at time t using only the observed prefix $X_{1:t}$ (Zhong et al., 2024; Shen & Elhamifar, 2024). Directly evaluating a full-sequence-trained model on successive prefixes causes severe over-segmentation and boundary displacement. Existing work mainly designs dedicated online architectures, uses progress or task priors, or aggregates observed history (Zhong et al., 2024; Shen & Elhamifar, 2024). We instead ask whether an existing offline segmentation backbone can be adapted through its training protocol.

We distinguish two sources of mismatch. Causal operations prevent a prediction from accessing future positions, but they do not determine which prefix lengths or endpoints occur during training. Full-video training concentrates observations at the completed-video state. Naive random truncation can also over-represent interiors of long actions and under-expose difficult states near transitions. We refer to this mismatch in observable input states as temporal context distribution shift (TCDS).

We introduce a training strategy that aligns online observation states without adding a test-time branch. Progress-balanced sampling distributes training prefixes across observation stages. Transition-guided refinement increases the probability that a prefix ends near an

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.

action change. Prefix-end weighting then increases classification weights toward the location whose prediction would be committed online. The method is evaluated with FACT on all official splits of GTEA and 50Salads and with progressive ablations on split 1. Cross-backbone experiments with ASFormer and MS-TCN clarify where the strategy transfers and where it does not.

Our contributions are threefold. First, we identify the mismatch between full-video training and strictly online prefix inference, and show why causal adaptation alone does not fully address it. Second, we propose a prefix-learning strategy that balances observation progress, targets action transitions, and weights supervision toward the prediction endpoint. Third, experiments across datasets and backbones show substantial gains over direct and causal online adaptation, while also exposing a limitation for a purely convolutional backbone.

2 Related Work

Temporal action segmentation. Fully supervised TAS learns action categories and temporal structure from frame-level annotations. Early approaches relied on temporal convolutional networks. The encoder–decoder TCN models action changes using one-dimensional convolutions (Lea et al., 2017), while MS-TCN stacks several full-resolution prediction stages and expands the receptive field with dilated convolutions (Farha & Gall, 2019). Later work explicitly addressed segment structure. ASRF combines action classification with class-agnostic boundary regression (Ishikawa et al., 2021), and RTK generates and refines sparse temporal keypoints (Jiang et al., 2023). These studies show that frame accuracy alone is insufficient because coherent segments and well-localized transitions are also required.

Attention provides a complementary means of capturing temporal relationships. ASFormer combines dilated temporal convolutions with local self-attention (Yi et al., 2021). UVAST formulates segmentation as sequence translation and restores frame labels through alignment (Behrmann et al., 2022). DiffAct treats the label sequence as the outcome of conditional denoising (Liu et al., 2023). FACT maintains frame representations and action tokens, then exchanges frame-level and segment-level information through cross-attention (Lu & Elhamifar, 2024). These models are generally optimized under complete-video protocols, which do not explicitly reproduce the changing prefix states encountered online.

Online and streaming segmentation. Offline TAS predicts after receiving a complete video. Online TAS must instead commit labels as frames arrive and form a continuous frame-level segmentation from those decisions. This requirement differs from online action recognition, whose primary output is the category of the current observation. It also differs from chunk-based streaming segmentation, where the model may inspect an entire short clip before producing its result.

Streaming TAS processes consecutive clips, sometimes conditioning on variable clip lengths (Wen et al., 2023; Zhang et al., 2025). Clip duration and within-clip visibility jointly affect latency and prediction quality. Strictly online methods commit the current prediction without future frames. ProTAS adapts segmentation models using causal operations, action progress, and task-graph constraints (Shen & Elhamifar, 2024). OnlineTAS continually aggregates observed history through adaptive memory and applies label-level post-processing to reduce fragmentation (Zhong et al., 2024). Other online approaches use efficient two-stream modeling or weakly supervised training (Kang et al., 2022; Ghoddoosian et al., 2022). Existing work therefore focuses mainly on online architectures, historical memory, causal modeling, and task priors. We complement these directions by controlling which prefix states enter training and how they are supervised.

3 Method

Existing TAS models commonly learn from complete videos, whereas strictly online inference receives a prefix that grows at every time step. The available history is limited near the beginning and gradually increases as the video unfolds. The model must consequently operate under a sequence of changing observation conditions rather than a single fixed input

regime. We address this discrepancy without introducing a new segmentation backbone. Our approach modifies the construction of training inputs and the distribution of frame-level supervision, while retaining the original segmentation objective and test-time network.

3.1 Temporal Context Distribution Shift

Let an untrimmed video of length T have features $X = (x_1, \ldots, x_T)$ and frame labels $Y = (y_1, \ldots, y_T)$ over class set $\mathcal{C}$. Offline and strictly online predictions at position t differ in their available context:

$$\hat{y}_t^{\text{off}} = \arg\max_{c \in \mathcal{C}} p_\theta(y_t = c \mid X_{1:T}), \qquad \hat{y}_t^{\text{on}} = \arg\max_{c \in \mathcal{C}} p_\theta(y_t = c \mid X_{1:t}). \tag{1}$$

Full-video training fixes the observation endpoint at T, whereas online inference encounters short, intermediate, and nearly complete prefixes. With relative progress $\rho_t = t/T$, the corresponding observation-state distributions satisfy

$$q_{\text{train}}(\rho) \neq q_{\text{online}}(\rho), \qquad 0 < \rho \leq 1. \tag{2}$$

Here, distribution refers to prefix length, observation progress, and endpoint location, rather than only numerical feature statistics. Causal convolution or masked attention restricts future access (Vaswani et al., 2017; Shen & Elhamifar, 2024), but does not by itself change the prefixes shown during training. We therefore combine causal adaptation with explicit construction and reweighting of training prefixes.

Figure 1 illustrates the distinction. A causal convolution restricts the receptive field through directed temporal connections, and causal self-attention masks keys located after the query. Both mechanisms answer whether an intermediate representation can aggregate future information. Prefix construction answers a different question: which observation ranges the network actually encounters during optimization. For models with global feature exchange, the latter can change intermediate representations even when future access is otherwise restricted.

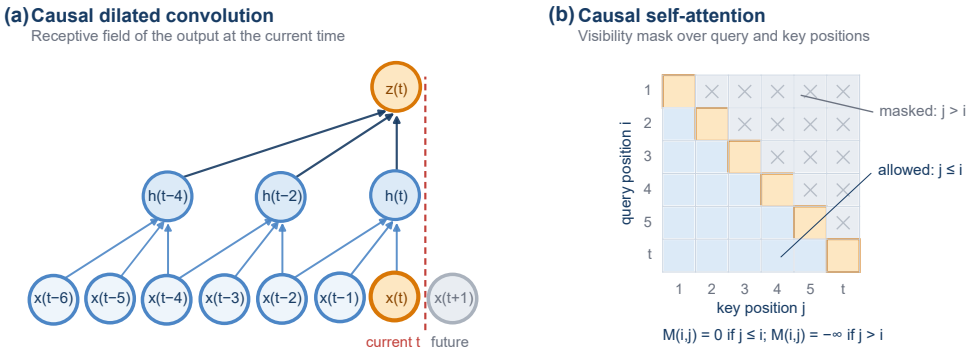

Figure 1: Causal temporal modeling. Directed convolutional connections and a lower-triangular attention mask restrict each representation to current and historical positions.

3.2 Temporal Context Distribution Alignment

Figure 2 summarizes the framework. Training begins from a labeled complete video. A sampler chooses endpoint t and jointly truncates features and labels to $(X_{1:t}, Y_{1:t})$. The observed prefix is processed by a causalized FACT backbone. FACT's frame and action branches retain their original cross-attention exchange within the visible prefix (Lu & Elhamifar, 2024). Frame classification is reweighted toward the endpoint, while the remaining FACT objectives are unchanged. At test time, the sampler and ground-truth transitions are removed. The learned model processes $X_{1:t}$ and commits the current prediction.

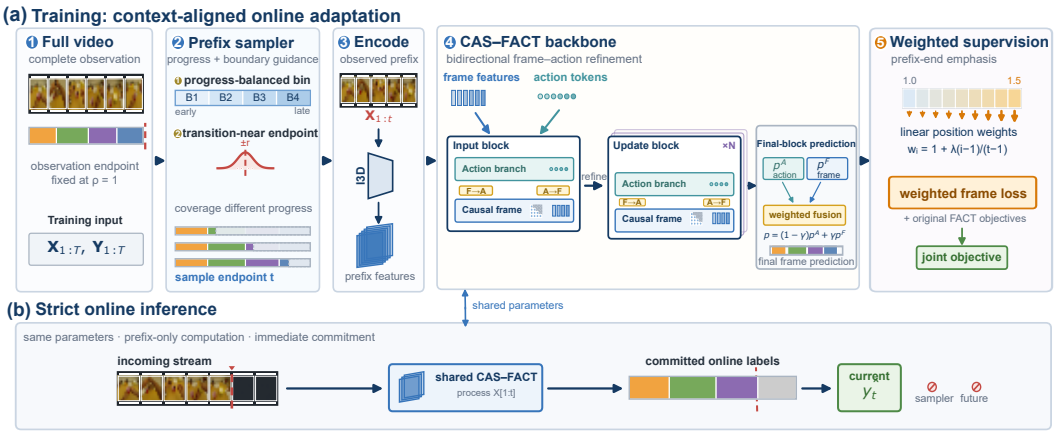

Figure 2: Temporal context distribution alignment. Training constructs progress-balanced and transition-guided prefixes, processes them with the causalized FACT backbone, and applies prefix-end-weighted supervision. Strictly online inference reuses the learned parameters and accesses only the observed prefix.

3.3 Progress- and Transition-Guided Prefix Sampling

Random endpoint selection generates incomplete inputs but does not control how often different observation stages or transition-neighborhood states appear. We first partition legal endpoints into K consecutive progress intervals. For minimum prefix length $L_{\min}$,

$$\{L_{\min}, \dots, T-1\} = I_1 \cup \cdots \cup I_K, \qquad I_i \cap I_j = \varnothing \ (i \neq j), \tag{3}$$

and select each interval with probability $P(k = j) = 1/K$. This produces approximately equal numbers of training prefixes at early, intermediate, and late observation stages, independent of absolute video duration.

The two-stage choice is important. Sampling an endpoint directly from all legal frames may appear neutral, but it provides no explicit control over the number of samples assigned to each observation stage. By selecting the progress interval first, each stage receives the same expected sampling opportunity. The endpoint is then chosen only within that interval, so videos of different durations are compared through relative progress rather than absolute frame indices.

We then use frame annotations to identify action transitions

$$\mathcal{B} = \{b \mid y_b \neq y_{b-1},\ 2 \leq b \leq T\}. \tag{4}$$

Within selected interval I_k, transition-guided sampling is invoked with probability p_b. An eligible b is selected and the endpoint is drawn uniformly from $I_k \cap [b-r, b+r]$, where r is the neighborhood radius. Otherwise, t is sampled uniformly from I_k. If no eligible transition intersects the interval, the procedure falls back to uniform interval sampling. Features and labels are truncated together after selecting t.

This refinement preserves the progress allocation established in the first stage. It changes only the endpoint distribution inside the selected interval. Consequently, transition-rich portions do not dominate the global prefix distribution, and videos without a transition in a particular interval still yield a valid training sample. Once t is obtained, the complete history $X_{1:t}$ is retained rather than replacing it with a fixed-width local window.

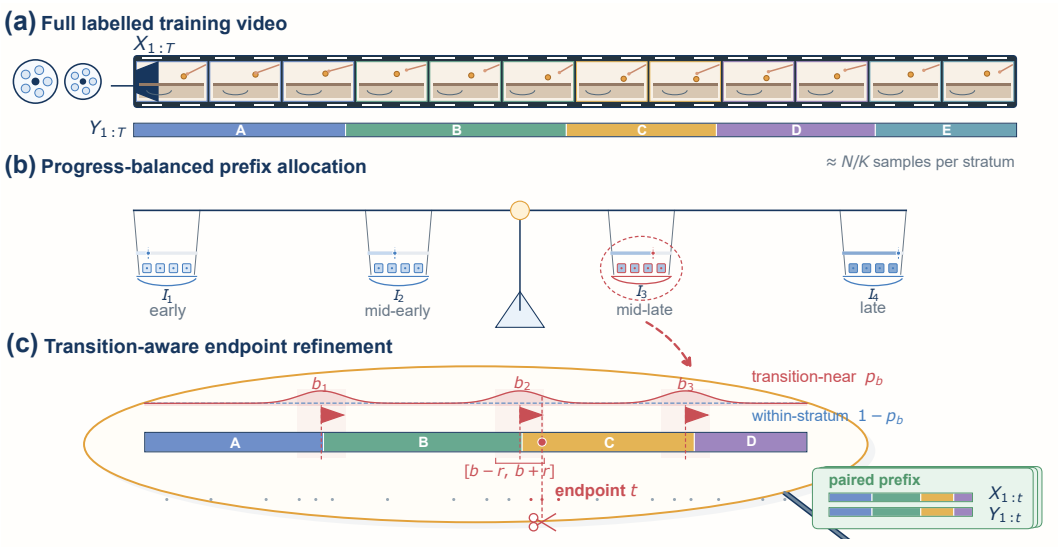

Figure 3: Progress-balanced and transition-guided prefix sampling. Candidate endpoints are first balanced across relative-progress intervals and then, with probability p_b, refined toward a neighborhood of a ground-truth action transition. Labels guide sampling only during training.

Progress balancing controls global coverage, whereas transition refinement increases exposure to local ambiguity, delayed label changes, and short actions near boundaries (Ishikawa et al., 2021; Jiang et al., 2023; Du et al., 2022). Importantly, boundary annotations are neither network inputs nor test-time requirements. The sampler therefore uses supervision only to decide where a training example ends. It neither supplies the transition index to the backbone nor changes the online inference protocol.

3.4 Prefix-End-Weighted Supervision

For a prefix of length t, conventional frame classification weights every observed position equally, although the endpoint is the prediction committed at the current online time. We retain supervision over the complete prefix and assign position i the weight

$$w_i = 1 + \lambda \frac{i-1}{t-1}, \qquad i = 1, \ldots, t. \tag{5}$$

The weight grows linearly from 1 to $1 + \lambda$. Let p_{i,y_i} be the predicted probability of the ground-truth class and α_{y_i} the original class weight. The weighted frame loss is

$$\mathcal{L}^{w}_{\text{frame}} = -\frac{1}{t} \sum_{i=1}^{t} w_i \alpha_{y_i} \log p_{i,y_i}. \tag{6}$$

The final objective replaces only FACT's frame classification term:

$$\mathcal{L} = \mathcal{L}^{w}_{\text{frame}} + \mathcal{L}_{\text{action}} + \mathcal{L}_{F \to A} + \mathcal{L}_{A \to F} + \lambda_s \mathcal{L}_{\text{smooth}}. \tag{7}$$

The action classification, bidirectional matching, and smoothing terms follow FACT (Lu & Elhamifar, 2024). Sampling changes which online states enter training. Endpoint weighting changes their within-prefix supervision. Neither adds inference computation.

The weighting is deliberately smooth rather than restricted to the final frame. Supervising only the endpoint would discard useful temporal structure and make optimization depend heavily on a single position. Linear weighting retains every observed label while increasing the relative contribution of recent positions. When $\lambda = 0$, the formulation reduces to the original equal-weight frame objective, which makes the modification directly comparable to the backbone loss.

4 Experiments

4.1 Setup

We evaluate on GTEA and 50Salads, two standard TAS datasets. GTEA contains 28 egocentric kitchen videos and 11 fine-grained actions, organized into four subject-based splits (Fathi et al., 2011). 50Salads contains 50 salad-preparation videos. We use its mid-level labels and five official splits (Stein & McKenna, 2013). We retain each backbone's public split definitions, input features, and class mappings, without changing the membership of the training or test sets.

We report frame accuracy (Acc), segmental edit score (Edit), and segmental F1 at IoU thresholds $\{0.1, 0.25, 0.5\}$ (Lea et al., 2017; Farha & Gall, 2019). Acc measures frame-level classification. Edit compares the predicted and ground-truth action orders after consecutive duplicate labels are collapsed. Segmental F1 matches predicted and ground-truth segments by temporal intersection over union and therefore penalizes fragmentation and boundary displacement. Unless explicitly marked as post-processed, all results are calculated from raw predictions.

FACT is trained with Adam (Kingma & Ba, 2015), learning rate 10^{-4}, batch size 1, and gradient-norm limit 10. We use $L_{\min} = 16$, $K = 4$, $p_b = 0.3$, $r = 20$, and $\lambda = 0.5$. ASFormer and MS-TCN retain their public optimization settings. Offline uses original full-video training and inference. Base applies an offline-trained model to successive prefixes. CAS combines causal architectural adaptation with ordinary prefix training. Ours adds endpoint weighting and progress/transition-guided sampling. Final FACT results are averaged over every official split. Progressive ablations use split 1 because successive-prefix evaluation is computationally expensive.

Strictly online evaluation recomputes a growing prefix and commits the current prediction at each step. This protocol is substantially more expensive than processing each complete video once. We therefore use two complementary evaluation levels. Final FACT and cross-backbone results are averaged over every official split, while detailed progressive ablations use split 1 of each dataset. The first level measures overall behavior across data partitions, and the second isolates how performance changes as the training strategy is assembled.

4.2 Main Results and Ablations

Table 1 shows that direct online inference sharply degrades segment quality. On GTEA, Base reaches only 25.97 F1@0.5, while CAS recovers 59.10 and Ours reaches 68.42. On 50Salads, Ours improves F1@0.5 from 12.82 for CAS to 39.68. Improvements occur across Acc, Edit, and all F1 thresholds, indicating better frame decisions and segment structure.

Table 1: FACT results averaged over all official splits (%).

Data	Setting	Acc	Edit	F1@.1	F1@.25	F1@.5
GTEA	Offline	83.77	89.62	91.98	90.60	80.27
	Base	33.86	47.20	48.78	41.26	25.97
	CAS	71.09	74.96	78.21	73.87	59.10
	Ours	76.41	79.20	82.47	79.33	68.42
50Salads	Offline	81.61	72.94	79.99	77.89	70.72
	Base	33.47	8.73	9.88	7.35	3.53
	CAS	44.42	18.21	21.32	17.82	12.82
	Ours	76.05	44.84	50.54	48.14	39.68

Table 2 reports progressive ablations. On GTEA, adding endpoint weighting to CAS mainly raises Acc, while BPB sampling increases Edit from 69.43 to 76.56 and F1@0.5 from 58.50 to 66.67. The slight reduction in Acc after adding BPB is accompanied by clear gains in every segment-oriented metric. This trade-off indicates that boundary-aware coverage primarily improves temporal organization rather than isolated frame decisions. On 50Salads, BPB

increases F1@0.5 from 18.35 to 39.59 and raises Acc to 77.85. Because the experiment is progressive, each row quantifies the next addition to the preceding configuration rather than the isolated effect of a module.

Table 2: Progressive FACT ablations on split 1 (%). B denotes prefix-end weighting. BPB denotes progress- and transition-guided sampling.

Data	CAS	B	BPB	Acc	Edit	F1@.1	F1@.25	F1@.5
GTEA				26.41	43.86	44.53	35.16	16.41
	✓			66.69	66.70	72.73	68.01	57.24
	✓	✓		75.31	69.43	73.47	70.75	58.50
	✓	✓	✓	72.92	76.56	77.04	74.81	66.67
50Salads				35.35	6.03	7.68	6.16	3.12
	✓			38.23	17.83	20.82	17.00	11.08
	✓	✓		57.87	23.69	27.93	25.28	18.35
	✓	✓	✓	77.85	43.27	53.47	51.10	39.59

4.3 Cross-Backbone and Published Comparisons

To test transfer beyond FACT, we apply the same training strategy to ASFormer. Table 3 gives complete results over all official splits. On GTEA, Ours improves Acc and F1@0.5 over CAS by 0.92 and 2.67 points, although Edit and the lower-threshold F1 scores decrease. The benefit therefore concerns stricter segment overlap rather than uniform improvement. On 50Salads, Edit increases by 8.75 points and F1@0.5 by 6.13 points, with gains at all F1 thresholds. Together with the FACT results, this confirms that the training strategy is not tied to FACT's frame–action representation alone.

Table 3: ASFormer results averaged over all official splits (%).

Data	Setting	Acc	Edit	F1@.1	F1@.25	F1@.5
GTEA	Offline	79.12	87.16	90.38	89.26	79.80
	Base	43.68	48.56	50.09	39.33	22.70
	CAS	74.22	78.54	83.82	81.35	68.68
	Ours	75.14	76.70	81.48	79.80	71.35
50Salads	Offline	87.04	81.33	87.11	85.25	79.73
	Base	50.48	8.23	10.61	8.82	5.93
	CAS	75.21	52.94	62.37	58.05	48.79
	Ours	74.54	61.69	69.88	65.65	54.92

We also evaluate the predominantly convolutional MS-TCN to identify the scope of transfer. Table 4 summarizes the comparison with CAS, while full metrics appear in Appendix A. Unlike FACT and ASFormer, MS-TCN does not benefit from the complete strategy. F1@0.5 decreases by 11.31 points on GTEA and 7.96 points on 50Salads. This negative result prevents a backbone-agnostic claim and documents that the adaptation depends on the temporal modeling mechanism.

Table 4: Compact MS-TCN cross-backbone result (%). Full results are in Appendix A.

Data	CAS Edit	Ours Edit	CAS F1@.5	Ours F1@.5
GTEA	63.92	56.78	56.17	44.86
50Salads	25.62	20.10	23.12	15.16

One plausible explanation lies in how the backbones aggregate temporal context. MS-TCN expands its receptive field through stacked dilated convolutions, so shorter training prefixes may reduce the historical context available near the endpoint. By contrast, attention

modules in FACT and ASFormer can dynamically connect positions within the observed prefix. These results suggest that prefix-based adaptation may be better suited to attention-based architectures that flexibly integrate visible information.

Table 5 positions the method against published strictly online results. Protocols, backbones, and features differ, so this is not a controlled estimate of improvement. Ours attains the strongest listed Edit and F1@0.5 without label-level post-processing on both datasets, while OnlineTAS has higher 50Salads Acc. Controlled evidence remains the FACT comparisons in Tables 1 and 2.

Table 5: Comparison with published strictly online TAS results without post-processing (%).

Data	Method	Acc	Edit	F1@.1	F1@.25	F1@.5
GTEA	LSTR (Xu et al., 2021)	63.7	33.2	41.5	37.7	25.0
	OODL (Ghoddoosian et al., 2022)	74.0	64.7	70.3	66.9	54.1
	OnlineTAS (Zhong et al., 2024)	76.2	63.5	72.6	68.3	58.8
	ProTAS (Shen & Elhamifar, 2024)	77.0	74.1	80.2	77.5	66.1
	BCFA (Han et al., 2025)	77.3	78.6	82.2	79.4	67.6
	Ours (FACT)	76.4	79.2	82.5	79.3	68.4
50Salads	LSTR (Xu et al., 2021)	60.5	5.0	8.2	6.6	4.1
	OnlineTAS (Zhong et al., 2024)	80.9	28.8	36.1	31.0	23.3
	Ours (FACT)	76.1	44.8	50.5	48.1	39.7

Post-processing analysis. Our main results are based on raw predictions, but strictly online outputs may still contain short fragments and boundary errors. We therefore examine how much label-level post-processing can correct these residual errors, using OnlineTAS as a contextual reference. Our GTEA rule performs short-range backward label filling at detected action onsets. On 50Salads, transition statistics estimated from the training set are used to suppress implausible short segments. These dataset-specific operations act only on predicted labels and are not part of the proposed training framework.

Table 6 summarizes the F1@0.5 changes. Here, p.p. denotes post-processing. Our F1@0.5 improves by 1.25 points on GTEA and 10.39 points on 50Salads. OnlineTAS gains more, particularly on 50Salads, where its raw output contains many fragmented short segments. Its correction rules therefore have greater opportunity to merge label fluctuations. The smaller gain for our method is consistent with its stronger raw segment continuity, although the comparison also depends on the dataset-specific correction rules. Complete metrics appear in Appendix C.

Table 6: Compact post-processing comparison in F1@0.5 (%). p.p. denotes post-processing.

Data	Method	Raw	+p.p.	Gain
GTEA	Ours	68.42	69.67	+1.25
	OnlineTAS	58.8	67.7	+8.9
50Salads	Ours	39.68	50.07	+10.39
	OnlineTAS	23.3	64.1	+40.8

Figure 4 provides a qualitative 50Salads example. Base produces many short segments and frequent class changes. Compared with CAS, Ours shows fewer spurious segments across the full sequence and better matches the ground-truth labels and boundaries in the highlighted interval.

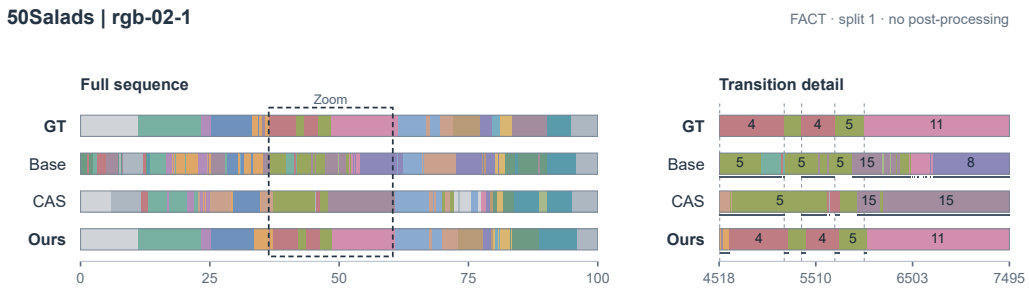

Figure 4: Strictly online segmentation on 50Salads. Rows show ground truth, Base, CAS, and Ours. The right panel enlarges the dashed interval. No post-processing is applied.

4.4 Discussion

The results indicate that adapting offline temporal action segmentation models to strictly online inputs requires more than restricting access to future frames. Aligning training prefixes and supervision with online inference improves FACT and also benefits ASFormer. The contrasting MS-TCN result suggests that the effectiveness of prefix-based adaptation is related to how a backbone aggregates temporal context.

The proposed sampling and weighting strategies operate only during training and introduce no additional inference branch. In the current evaluation, each growing prefix is processed independently so that the prediction at each time step uses only observed frames. This implementation repeats computation over earlier frames. This cost arises from independently recomputing prefixes in the current evaluation, rather than from the proposed training strategy.

5 Conclusion

We presented a training-time strategy for adapting full-sequence temporal action segmentation models to strictly online inputs by aligning prefix exposure and supervision with evolving observations. On GTEA and 50Salads, the adapted FACT model improves F1@0.5 over causal adaptation by 9.32 and 26.86 percentage points, respectively. ASFormer also shows gains in F1@0.5, suggesting that this training principle extends beyond the primary backbone among the attention-based models evaluated. Together, these findings identify alignment of training prefixes and online observations as an important complement to causal modeling.

AI Use Statement

Generative AI tools were used for literature organization, language editing, and conversion of the manuscript to the ICLR template. The authors verified all citations, technical descriptions, analyses, and revisions, and take responsibility for the final content of this work.

References

Nadine Behrmann, S. Alireza Golestaneh, Zico Kolter, Jürgen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In ECCV, 2022.

Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. TPAMI, 46(2):1011–1030, 2024.

Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In CVPR, 2022.

Yazan Abu Farha and Jürgen 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.

Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, Chiho Choi, and Behzad Dariush. Weakly-supervised online action segmentation in multi-view instructional videos. In CVPR, 2022.

Zhangye Han, Xun Jiang, Zheng Wang, Xin Liu, Fumin Shen, and Xing Xu. Egocentric online action segmentation with behavior-centred feature augmentation. In ICME, 2025.

Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In WACV, 2021.

Borui Jiang, Yang Jin, Zhentao Tan, and Yadong Mu. Video action segmentation via contextually refined temporal keypoints. In ICCV, 2023.

Min-Seok Kang, Rae-Hong Park, and Hyung-Min Park. Efficient two-stream network for online video action segmentation. IEEE Access, 10:90635–90646, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

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

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.

Yuhan Shen and Ehsan Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. In CVPR, 2024.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, 2017.

Wujun Wen, Yunheng Li, Zhuben Dong, Lin Feng, Wanxiao Yang, and Shenlan Liu. Streaming video temporal action segmentation in real time. In International Conference on Intelligent Systems and Knowledge Engineering, 2023.

Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In NeurIPS, 2021.

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

Jinrong Zhang, Wenjun Wen, Shenglan Liu, Sifan Zhang, Yuning Ding, and Lin Feng. Flexible streaming temporal action segmentation with diffusion models. In ICME, 2025.

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

A Complete MS-TCN Results

Table 7: MS-TCN results averaged over all official splits (%).

Data	Setting	Acc	Edit	F1@.1	F1@.25	F1@.5
GTEA	Offline	79.26	83.10	87.85	86.58	73.06
	Base	43.39	57.34	64.00	54.20	37.98
	CAS	74.03	63.92	70.50	68.08	56.17
	Ours	65.40	56.78	62.91	58.95	44.86
50Salads	Offline	70.75	59.11	67.46	64.09	53.98
	Base	53.57	34.60	43.24	37.53	22.82
	CAS	68.86	25.62	33.01	30.23	23.12
	Ours	57.85	20.10	25.06	22.79	15.16

B Additional Qualitative Result

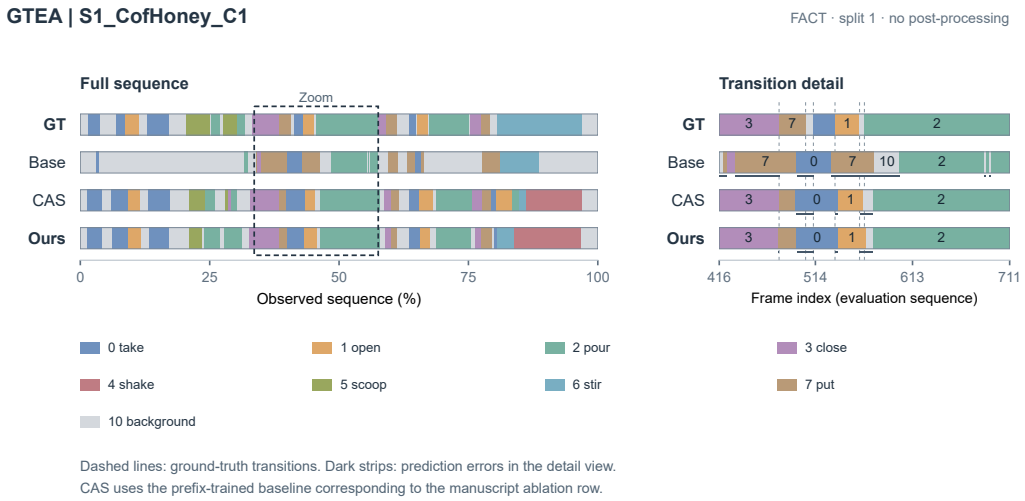

Figure 5: Strictly online segmentation on GTEA. Rows show ground truth, Base, CAS, and Ours. No post-processing is applied.

C Supplementary Post-Processing

Our post-processing operates only on predicted labels. GTEA uses short-range backward label filling at action onsets. 50Salads suppresses implausible short segments using transitions estimated from the training set. These dataset-specific rules are not part of the unified training method.

Table 8: Results before and after post-processing (p.p), in %.

Data	Setting	Acc	Edit	F1@.1	F1@.25	F1@.5	ΔF1@.5
GTEA	Ours	76.4	79.2	82.5	79.3	68.4	–
	Ours+p.p	73.0	80.1	82.9	79.9	69.7	+1.3
	OnlineTAS	76.2	63.5	72.6	68.3	58.8	–
	OnlineTAS+p.p	74.4	70.3	78.5	76.4	67.7	+8.9
50Salads	Ours	76.1	44.8	50.5	48.1	39.7	–
	Ours+p.p	75.6	56.3	62.7	60.3	50.1	+10.4
	OnlineTAS	80.9	28.8	36.1	31.0	23.3	–
	OnlineTAS+p.p	77.7	71.5	77.7	74.6	64.1	+40.8