

ASO-Temba: Adaptive-Scale Online Mamba for Understanding Long Untrimmed Videos

Niccolò Pasi

3194064@studbocconi.it

Raffaele Simbari

3185670@studbocconi.it

Samuele Nicolò Straccialini

3199302@studbocconi.it

Bocconi University

Abstract

*Temporal Action Detection (TAD) on long, densely labeled, videos of Activities of Daily Living (ADL) requires models that simultaneously capture fine-grained boundaries and very long temporal context, as well as ideally producing predictions as the video streams in. MS-Temba, a recent Mamba-based architecture, addressed the first requirement through dilated state-space blocks and a multi-scale Mamba fuser, but (i) treats the fusion of multi-scale features as a fixed SSM aggregation and (ii) is fundamentally offline, relying on bidirectional scans and full-video context. We revisit MS-Temba along these two axes. First, we **replaced the Mamba fuser with some alternatives**, such as a learned weighted sum and some attention-based variants, and study how multi-scale temporal features are best combined for dense ADL detection. Second, we introduce a **fully causal, streaming** variant of MS-Temba: all SSMs are made forward-only, the fuser is causalized, and a new causal-consistency loss replaces the original C-state diversity loss. On Toyota Smarthome Untrimmed (TSU) and Charades, our attention-based fusers show improvements with respect to the offline baseline, and the causal streaming model retains most of offline accuracy while enabling online TAD with zero look-ahead. A complete implementation can be found in the following repository: <https://github.com/sstraccialini/ASO-Temba>.*

1. Introduction

Long-form Temporal Action Detection (TAD) on untrimmed videos of Activities of Daily Living (ADL) is one of the hardest settings of video understanding. Compared with standard TAD benchmarks, ADL videos are (i) much longer (spanning tens of minutes or even hours), (ii) *densely* annotated, with many co-occurring and partially overlapping actions, and (iii) characterized by subtle, low-motion classes that nevertheless require sharp temporal boundaries. In addition, many real-world

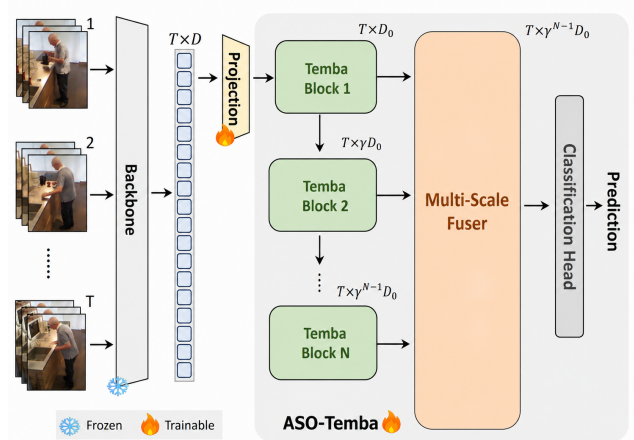


Figure 1. **Overview.** ASO-Temba architecture is based on the MS-Temba one [12]: dilated SSM blocks are stacked at multiple temporal scales and then fused them into per-frame action probabilities. We first revisit the multi-scale fuser, replaced by weighted and attention-based variants, then make the architecture causal, where all SSMs are made forward-only.

applications, such as elderly monitoring, and assistive robotics, require the detector to operate *online*. Therefore predictions must be produced as the video streams in, without access to future frames and under limited latency constraints.

Recent State-Space Models (SSMs), and in particular Mamba [7], have emerged as a compelling alternative to Transformers for very long-range sequence modeling, enabling efficient processing of long sequences. MS-Temba [12] adapts Mamba to TAD by combining multiple dilated *dilated* SSM branches to capture multi-scale temporal dynamics at different receptive scales, aggregating their outputs with a lightweight Multi-Scale fusion stage. With only $\sim 17\text{M}$ parameters it reaches state-of-the-art performance on TSU and Charades.

However, two aspects of MS-Temba remain unaddressed. First, the fuser is not adaptive: Multi-scale out-

puts are merged through a fixed aggregation rule, with no explicit mechanism to adapt their relative importance to the current temporal context. Second, MS-Temba is inherently offline: the dilated blocks use bidirectional Mamba scans, the fuser sees the full sequence, and the C-state diversity loss compares states across the whole video. None of these is compatible with online streaming inference.

Contributions. In this work, we extend MS-Temba along these two independent directions:

1. **Fuser redesign (§3.2).** We replace the original fusion module with several alternatives including a per-scale learned *weighted* sum, and multiple *attention*-based fusers such as *token-attention*, *cross-token-attention*, a full *attention* fuser and *AttentionX3*. These variants share the same input-output interface, allowing a clean comparison of multiple fusion mechanisms.
2. **Causal streaming MS-Temba (§3.5).** We convert the whole pipeline to be *causal*: forward-only Mamba scans, a causal multi-scale fuser, and a new *causal-consistency* loss $\mathcal{L}_{\text{caus-cons}}$ that replaces the offline C-state diversity loss while preserving its role of decorrelating scale-specific branches.

2. Related Work

Temporal Action Detection. In activity-of-daily-living benchmarks such as Charades and Toyota Smarthome Untrimmed (TSU), actions may overlap, last for very different durations, and require both fine temporal boundaries and long-range temporal context. Therefore, earlier TAD methods moved beyond single-scale temporal modeling. PDAN [3] uses pyramid dilated temporal attention to capture dependencies at multiple temporal scales, while MS-TCT [4] combines temporal convolutions and Transformer-style modeling in a multi-scale hierarchy. More general temporal localization models such as ActionFormer [14] further exploit multi-scale temporal representations.

State-space models and Mamba. Recent State Space Models (SSMs) have emerged as an efficient alternative to Transformers for long-range sequence modeling. In particular, Mamba [7] introduced selective SSMs with linear-time scans, and several recent works [8, 12] apply Mamba to video.

Online action detection. A separate line of work targets *online* action detection, in which the model must commit to a prediction at frame t using only past context [6, 13]. Most of these approaches use transformer decoders with causal masking or recurrent memory modules. Unlike these, our streaming MS-Temba is built on the multi-scale dilated SSM design of MS-Temba.

Multi-scale feature fusion. Multi-scale aggregation is a central topic of interest in dense prediction. Feature Pyra-

mid Networks [9] popularized the idea of combining features at different resolutions for object detection, and temporal action-detection models similarly rely on multi-scale temporal representations to capture both short actions and long activities. In this paper, we study adaptive alternatives to the original MS-Temba fuser to preserve the same input-output interface, but make the relative contribution of each scale explicit.

3. Methodology

3.1. Background: MS-Temba

Following MS-Temba, an untrimmed video is pre-processed into a sequence of snippet features $X \in \mathbb{R}^{T \times D}$ using a frozen backbone (I3D or CLIP). A stack of K *Temba blocks* processes X ; each block is a bidirectional Mamba whose internal 1D convolution and SSM are run with a different *dilation* d_k , exposing a different temporal receptive field. Let $Y_k \in \mathbb{R}^{T \times D}$ denote the output of the k -th block. The Multi-Scale Mamba Fuser $\mathcal{F}$ aggregates $\{Y_1, \dots, Y_K\}$ into a single sequence $Z = \mathcal{F}(Y_1, \dots, Y_K) \in \mathbb{R}^{T \times D}$, which is fed to a per-frame multi-label classifier $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^C$ producing $\hat{p}_t = \sigma(\phi(Z_t))$. Training uses a binary cross-entropy loss $\mathcal{L}_{\text{bce}}$, an auxiliary per-block classification loss $\mathcal{L}_{\text{aux}}$, and a C-state diversity loss $\mathcal{L}_{\text{div}}$ that encourages the SSM states of different dilation branches to be decorrelated. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{bce}} + \alpha \mathcal{L}_{\text{aux}} + \beta \mathcal{L}_{\text{div}}. \quad (1)$$

3.2. Multi-scale fuser variants

The original fuser $\mathcal{F}$ is itself a Mamba: the per-scale outputs are concatenated along the time axis into a sequence of length $K \cdot T$ and processed by another selective SSM, which is impossible to causalize cleanly (§3.5). We replace $\mathcal{F}$ with simpler, scale-aware aggregators sharing the input signature $\{Y_k\}_{k=1}^K \rightarrow Z$: (§3.5).

(i) **Sum (baseline).** $Z_t = \sum_k Y_{k,t}$, equivalent to removing the fuser. Used as a reference point.

(ii) **Weighted.** A learned scalar (or per-channel) gate per scale, $Z_t = \sum_k w_k \odot Y_{k,t}$ with $w_k \in \mathbb{R}^D$. This is the cheapest non-trivial fuser and makes the inductive bias of different scales having different importance explicit.

(iii) **Token-attention.** At each timestep t , the K scale tokens $\{Y_{k,t}\}_k$ are treated as a K -long sequence and attended over, with $Z_t = \text{Attn}_t(Y_{1,t}, \dots, Y_{K,t})$. There is no temporal attention; the cost is $O(TK^2D)$.

(iv) **Cross-token-attention.** A query token aggregates the same K scale tokens via cross-attention; this is the variant of (iii) that explicitly decouples the query subspace from the key/value subspace and lets us control the output dimensionality.

(v) **Attention fuser.** A full attention-based fusion block combines projection, self-attention, feed-forward refinement, and dynamic scale routing. We describe it in detail in §3.3.

(vi) **Attention X3.** A structured extension of the attention fuser that preserves branch specialization through independent per-scale attention processing before fusion. Full details are given in §3.4.

While in the next paragraph we detail fusers (v) and (vi), we won’t include further discussion on the other methods mentioned above since they showed poor results with respect to the baseline, as evident from Table 1.

3.3. Attention fuser with dynamic routing

The variants in §3.2 either operate token-wise across scales or rely on fixed aggregation rules, with no explicit routing over the original scale representations. We therefore introduce an `attention` fuser that couples a self-attention contextualization block with a dynamic routing stage over the original per-scale features, raising the parameter count from $\sim 16.5\text{M}$ to $\sim 21.5\text{M}$.

Architecture. Given aligned scale outputs $\{Y_k\}_{k=1}^K$, with $Y_k \in \mathbb{R}^{B \times T \times D}$, the `attention` fuser proceeds in three stages.

(1) *Concatenation and projection.* The K scale features are concatenated along the channel dimension and projected back to D through a learned affine map followed by GELU:

$$\mathbf{x} = \text{GELU}(W_{\text{proj}}[Y_1 \parallel Y_2 \parallel \dots \parallel Y_K] + b_{\text{proj}}). \quad (2)$$

This step compresses the concatenated KD -dimensional multi-scale representation into a shared latent space suitable for contextual modeling.

(2) *Temporal self-attention and FFN.* The projected representation is processed by a standard pre-LayerNorm Transformer sub-block:

$$\mathbf{x} \leftarrow \mathbf{x} + \text{Dropout}(\text{MHA}(\text{LN}(\mathbf{x}))), \quad (3)$$

$$\mathbf{x} \leftarrow \mathbf{x} + \text{FFN}(\text{LN}(\mathbf{x})), \quad (4)$$

where FFN is a two-layer MLP with hidden dimension $4D$ and GELU activation. This sub-block injects global temporal context into each frame token, producing a contextual representation $\mathbf{x}$ used to guide routing across scales.

(3) *Dynamic routing.* Rather than using $\mathbf{x}$ directly as the fused output, we use it as a context query to compute per-frame routing weights over the original scale-specific features:

$$\mathbf{r}_t = \text{softmax}\left(\frac{W_r \text{LN}(\mathbf{x}_t) + b_r}{\tau}\right) \in \Delta^{K-1}, \quad (5)$$

where $\tau > 0$ is a learned temperature parameter, initialized to 1 and clamped to a minimum of 0.1, and $W_r \in \mathbb{R}^{K \times D}$ is

initialized to zero so that routing starts from a uniform distribution. The routed combination is then added residually to the contextual branch and layer-normalized:

$$Z_t = \text{LN}\left(\mathbf{x}_t + \sum_{k=1}^K r_{t,k} Y_{k,t}\right). \quad (6)$$

Training objective for the attention fuser. For the `attention` fuser, routing weights are learned jointly with the base detection objective. The overall loss is

$$\mathcal{L}_{\text{attn}} = \alpha_l \left(\mathcal{L}_f + \lambda_{\text{blk}} \sum_{i=1}^3 \mathcal{L}_{\text{blk}}^{(i)} \right) + \lambda_{\text{div}} \mathcal{L}_{\text{div}} - \lambda_{\text{ent}} \mathcal{H}(\mathbf{r}), \quad (7)$$

where $\mathcal{L}_f$ is the final frame-level binary cross-entropy loss, $\mathcal{L}_{\text{blk}}^{(i)}$ are the auxiliary block-level losses, and $\mathcal{L}_{\text{div}}$ is the diversity regularizer inherited from the base MS-Temba objective. The routing entropy is defined as

$$\mathcal{H}(\mathbf{r}) = -\frac{1}{BT} \sum_{b=1}^B \sum_{t=1}^T \sum_{k=1}^K r_{b,t,k} \log r_{b,t,k}, \quad (8)$$

with $r_{b,t,k}$ the routing weight assigned to scale k at time t for sample b . Since the entropy term is subtracted in Eq. 7, optimization encourages higher-entropy routing distributions, helping prevent premature router collapse onto a single branch and promoting more flexible multi-scale usage.

Architectural motivation. The `attention` fuser follows a deliberate two-pathway design. The Transformer sub-block (Eqs. 3–4) builds a contextual representation with global temporal information, while the routing stage (Eqs. 5–6) uses that representation to adaptively re-weight the original, unmodified scale features. The residual combination in Eq. 6 ensures that the fused representation retains direct access to explicit scale-specific information, instead of relying only on the transformed latent state $\mathbf{x}$.

3.4. AttentionX3: per-scale self-attention fuser

The multiscale attention fuser applies a dedicated self-attention sub-block to each Temba block output $z_k \in \mathbb{R}^{B \times T \times D_k}$ in its native embedding space, before any cross-scale mixing:

$$z'_k = \text{MHA}_k(\text{LN}(z_k)) + z_k \quad (9)$$

Figure 3 illustrates the full pipeline. Each z'_k is projected to $E=576$, summed across scales, and layer-normalized. Heads are set to $\lfloor D_k/64 \rfloor$, keeping per-head dimension fixed at 64. Inside each Temba block, dilated SSM branches process disjoint subsequences – branch i in block k only sees positions $\{i, i+k, i+2k, \dots\}$ – so the reassembled z_k contains tokens from branches that have never interacted,

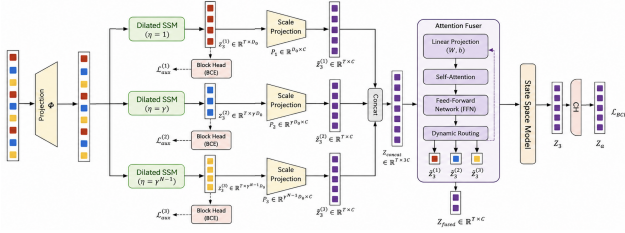


Figure 2. **Attention fuser.** Each Temba block output z_k is mapped to a shared dimension through a scale projection layer. The projected features are concatenated and re-encoded by a joint linear projection. Then they are processed by a self-attention block followed by a FFN. The resulting contextual representation drives a dynamic routing module. The fused output is then refined by the final Fuser SSM and passed to the classification head.

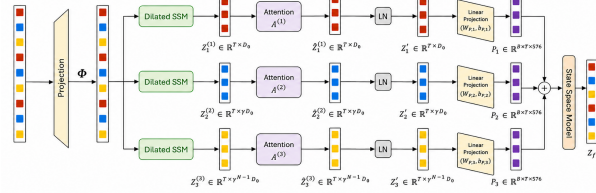


Figure 3. **AttentionX3 fuser.** Each Temba block output z_k is refined by a dedicated self-attention sub-block with residual connection, followed by layer normalisation, all operating in the native embedding space D_k . The three outputs are then projected to $E=576$, summed, and processed by the Fuser SSM.

and the sum fuser collapses these without any temporal routing. Self-attention before projection lets every frame attend to every other in the reassembled sequence, recovering the within-scale dependencies the dilated structure had broken. This is most consequential at block 3: its stride-3 scan compresses more temporal context into each SSM state, making z_3 the most informationally dense of the three outputs; with 9 heads in $D_3=576$, the multiscale attention fuser recovers local coherence and captures long-range dependencies before scales are mixed.

3.5. Causal streaming MS-Temba

Two components of MS-Temba are not causal: the bidirectional Mamba inside each Temba block and the diversity loss $\mathcal{L}_{\text{div}}$, which compares C-states accumulated over time.

Forward-only blocks. Each Temba block is restricted to a single forward Mamba scan; the backward branch, which was used to analyze the frames in reverse, is dropped. This preserves the multi-scale dilated structure but ensures that the output at time t depends only on inputs at times $\leq t$. At the same time, it reduces the representation power of the bidirectional Temba blocks. As discussed in §5, the accu-

racy gap remains limited.

Causal-consistency loss. $\mathcal{L}_{\text{div}}$ relies on full-video C-states and cannot be evaluated online. We replace it by a margin-based hinge loss on the per-timestep branch outputs:

$$\mathcal{L}_{\text{caus-cons}} = \frac{1}{T} \sum_{t=1}^T \sum_{i < j} \max(0, 1 - \cos(y_{t,i}, y_{t,j}) - m), \quad (10)$$

where $y_{t,k}$ is the output of branch k at time t and m is a margin (we default it at $m=0.1$), used to prevent branch representations from collapsing to the same point. $\mathcal{L}_{\text{caus-cons}}$ uses only information available at time t , so it can be computed in the streaming forward; it encourages scale branches to agree where they are confident, but stops penalizing them once they are already similar. The total causal objective is then simply given by equation 1, where we substitute $\mathcal{L}_{\text{div}}$ with $\mathcal{L}_{\text{caus-cons}}$:

$$\mathcal{L}^{\text{caus}} = \mathcal{L}_{\text{bce}} + \alpha \mathcal{L}_{\text{aux}} + \gamma \mathcal{L}_{\text{caus-cons}}. \quad (11)$$

Once the model is causal, the output at time t is a deterministic function of $X_{\leq t}$. We therefore implement two forward modes that are numerically equivalent by construction: a fast *parallel* scan over the full T -long sequence, used to compute quickly mAP at training and evaluation time, and a true *streaming* forward that consumes the video chunk-by-chunk (chunk size = 1 is frame-by-frame; for $N>1$ is an N -frame buffer) and maintains the SSM hidden states across calls. The last is useful especially to measure per-frame latency and real-time factor, avoiding the cost of running the model over the full test set but measuring potential performances in concrete application.

Causal fuser. The original Mamba fuser clearly fits our architecture with no need for modification. In line with the research presented in this paper, we can try modifying the fuser with a causal version of one of the attention fusers (§3.2). In particular, the token-attention fuser is naturally causal since it involves no temporal interaction, so we tested its performance on our causal architecture, hoping it could recover meaningful interactions between current and past tokens.

4. Datasets

We evaluate on the two densely labeled, long-form ADL benchmarks used by MS-Temba.

Toyota Smarthome Untrimmed (TSU) [5]. TSU contains 536 untrimmed videos of daily indoor activities and comprises 51 fine-grained and often overlapping action classes, with videos spanning several minutes and densely annotated at the frame level. In our setup, sequences are padded to 2500 frames, making TSU a particularly challenging benchmark for long-range temporal modeling, since semantically related actions may be separated by hundreds of frames.

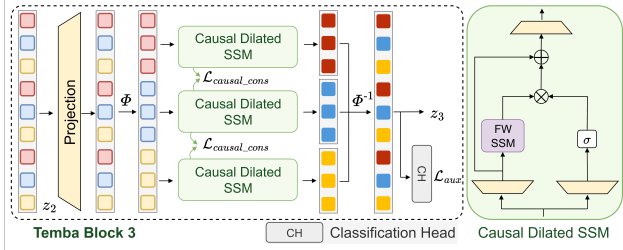


Figure 4. **Causal streaming MS-Temba.** Bidirectional Mamba scans are replaced by forward-only scans, and the C-state diversity loss is replaced by the causal-consistency loss $\mathcal{L}_{\text{caus-cons}}$ which only uses information up to time t .

Charades [11]. Charades consists of 9,848 home videos annotated with 157 action classes; in our experiments, we use the 24 fps version. Compared to TSU, videos are much shorter, averaging about 40 seconds and padded to 256 frames, but the action vocabulary is substantially larger and highly fine-grained. As a result, the main challenge is not long-range temporal memory, but accurate discrimination among short, overlapping domestic actions.

Features. We run our model on pre-extracted snippet features rather than raw frames. To have a direct match with the original MS-Temba paper, we use features extracted from two backbones: I3D [2], which captures motion-aware spatiotemporal dynamics, and CLIP [10], which provides richer semantic and contextual visual representations.

5. Results

We report per-frame mean Average Precision (mAP), the standard metric for dense multi-label TAD, in Table 1.

The simple sum is a strong baseline, indicating that most of the multi-scale signal is already captured by the dilated Temba blocks.

The **Weighted Sum** fuser does not improve over the static-sum baseline, and in fact slightly underperforms it in three out of four settings, with only a modest gain on Charades-CLIP. This suggests that assigning fixed global weights to the three scales is not sufficient to make fusion more effective, and that the main gains instead come from input-dependent aggregation.

The proposed **Attention** fuser consistently improves over the original static-sum baseline in all four settings. Relative to simpler attention mechanisms such as *Token-attention* and *Cross-token-attention*, the full attention-based design is clearly more effective, showing that contextualizing the fused representation before aggregation is substantially more beneficial than lighter token-level interaction. Although this design increases the parameter count by approximately 5M parameters, the resulting gains indicate that the additional capacity is used productively, espe-

Fuser	TSU		Charades	
	I3D	CLIP	I3D	CLIP
Sum (baseline)	36.45	43.62	24.74	31.95
Weighted	36.08	43.59	24.70	32.40
Token-attention	36.27	42.32	24.00	31.77
Cross-token-attention	34.20	42.89	24.25	30.91
Attention	<u>38.79</u>	45.83	<u>25.46</u>	<u>34.92</u>
AttentionX3	38.99	<u>44.97</u>	26.33	35.43

Table 1. **Results with different fusers (mAP, %).** *Attention* and *AttentionX3* fusers obtain higher mAP than other methods. Best per column in **bold**, second best underlined.

Method	TSU		Charades	
	I3D	CLIP	I3D	CLIP
MS-Temba (original offline)	36.45	43.62	24.74	31.95
MiniROAD (baseline)	34.65	42.71	21.79	29.25
ASO-Temba + token-attention	33.16	40.36	22.25	30.80
ASO-Temba (ours)	33.37	40.89	22.69	30.98

Table 2. **Results of causal architecture (mAP, %).** Our online model with standard fuser losses just 1-3% in accuracy with respect to the original offline architecture, outperforming the MiniROAD baseline on Charades.

cially in the TSU-CLIP setting, where *Attention* achieves the best overall result with a +2.21 mAP improvement over the baseline. Table 3 further analyzes the contribution of the two sub-blocks of the multiscale attention fuser.

Among the attention-based variants, **AttentionX3** achieves the best overall performance, ranking first on TSU-I3D, Charades-I3D, and Charades-CLIP, with gains of +2.54, +1.59, and +3.48 mAP over the baseline, respectively. Its advantage is particularly evident on Charades, where actions are dense and heavily overlapping. In this setting, processing each scale separately helps preserve the specialization of the dilated branches. Each scale is refined before fusion, instead of being collapsed into a shared representation too early. Relative to *Attention*, *AttentionX3* provides a modest but consistent performance gain, at the expense of approximately 1.7M additional parameters.

The **online** architecture results, summarized in Table 2, show that MS-Temba can be converted into a genuinely streaming architecture without collapsing in performance. Most of the accuracy of the offline model is preserved, with a drop of just 1-3% in mAP on both datasets and backbones. In particular, the causal model remains competitive on TSU, while it outperforms the MiniROAD¹ online

¹MiniROAD is a lightweight recurrent model, often used as baseline for online action detection [1]; since the original work does not report results on TSU and Charades under our feature setup, we adapt and train it using

Concat. + Proj.	Self Attention	FFN	Routing	TSU	Charades
✗	✗	✗	✗	43.62	31.95
✓	✗	✗	✗	45.00	34.14
✓	✓	✗	✗	45.82	33.70
✓	✓	✓	✗	44.17	33.10
✓	✓	✗	✓	45.11	34.75
✓	✓	✓	✓	45.83	34.92

Table 3. **Ablation study on CLIP of the attention-based fuser (mAP, %).** We progressively enable concatenation with projection, self-attention, feed-forward refinement, and dynamic routing.

baseline on Charades, suggesting that the multi-scale SSM structure remains effective even under strict causality constraints. The larger accuracy drop on TSU likely reflects its longer videos, where the lost backward context is more consequential than on Charades, where actions are short and densely overlapping. The main trade-off is the expected loss of future context, which affects accuracy especially when action boundaries or subtle activities require evidence from subsequent frames. Nevertheless, the relatively small performance gap indicates that forward-only multi-scale state-space modeling is a promising direction for real-time ADL understanding. Token-attention fuser, which we attempted as a simple causal modification since it involves no temporal interaction, did not show any improvement over the standard fuser.

5.1. Ablation: Attention

Table 3 reports the progressive contribution of the main components of the *Attention* fuser relative to the static-sum baseline. On both TSU and Charades (with CLIP backbone), simply replacing the static fusion rule with *concatenation + projection* already yields a noticeable improvement. This suggests that part of the gain comes from moving fusion from a fixed additive rule to a learned shared latent space, where the multi-scale representations can be jointly re-encoded before aggregation. Adding self-attention produces a further improvement on TSU, while leading to a slight decrease on Charades, indicating that the benefit of global temporal contextualization is not uniform across datasets. A more interesting pattern emerges when considering the FFN and the routing module. Adding the FFN alone after self-attention decreases performance on both CLIP-based settings, and in fact performs worse than the simpler *concat + proj + self-attention* variant. This shows that nonlinear channel mixing, when applied in isolation, does not by itself yield a better fused representation. By contrast, replacing the FFN with dynamic routing improves performance relative to the FFN-only variant,

the same I3D/CLIP inputs, label format, and evaluation protocol used for ASO-Temba.

suggesting that explicitly re-weighting the original scale-aligned features is more effective than refining the shared latent state alone. Most importantly, the strongest overall result is obtained when *both* FFN and dynamic routing are present. This indicates that the two components are complementary: the FFN enriches the contextual representation produced after self-attention, while the routing module uses that representation to drive adaptive scale selection. In this sense, the FFN is most useful not as a standalone refinement stage, but as a contextual pre-processing step for routing.

6. Conclusions and Future Work

In this work we revisited MS-Temba in two directions: *how* multi-scale temporal features are fused, and *when* predictions can be produced. On the fuser side, simple weighted and basic token-level attention aggregators fail to improve the original SSM fuser. However, more structured attention versions are able to increase the accuracy of the architecture; this comes at the cost of an additional 5–8M parameters. On the temporal side, a fully causal MS-Temba, featuring forward-only scans and causal-consistency loss, enables true online TAD with zero look-ahead while retaining most of the offline accuracy on both the TSU and Charades datasets. The two contributions are complementary: combining the best causal fuser with the streaming training recipe is a natural next step, as is end-to-end training from raw frames.

Future work may include reducing the parameter overhead of the fusion modules, combining causal attention-based fusers with the causal streaming training recipe, and evaluating the full online pipeline end-to-end including feature extraction from raw video.

Limitations

Our results show that adaptive attention-based fusion can improve over the original MS-Temba fuser, but this comes at the cost of additional parameters and computation. In particular, the best-performing fusers increase the model size by several million parameters, which may be undesirable in low-resource or real-time deployments. Moreover, the online version is evaluated on pre-extracted I3D and CLIP features, so its real-time factor only reflects the temporal model and does not include the cost of visual feature extraction from raw frames. Finally, the causal model necessarily loses access to future context, which can hurt predictions near action boundaries or for subtle activities that are easier to recognize after observing subsequent frames.

References

- [1] Jounghbin An, Hyolim Kang, Su Ho Han, Ming-Hsuan Yang, and Seon Joo Kim. Miniroad: Minimal rnn framework

for online action detection. In *International Conference on Computer Vision (ICCV)*, 2023. 5

- [2] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, 2018. arXiv:1705.07750. 5
- [3] Rui Dai, Srijan Das, Luca Minciullo, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Pdan: Pyramid dilated attention network for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2970–2979, 2021. 2
- [4] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael Ryoo, and Francois Bremond. MS-TCT: Multi-Scale Temporal ConvTransformer for Action Detection. In *CVPR*, 2022. 2
- [5] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 4
- [6] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection, 2016. 2
- [7] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, 2024. arXiv:2312.00752. 1, 2
- [8] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding, 2024. 2
- [9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 2
- [10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021. arXiv:2103.00020. 5
- [11] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ivan Laptev, Ali Farhadi, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv e-prints*, 2016. 5
- [12] Arkaprava Sinha, Monish Soundar Raj, Pu Wang, Ahmed Helmy, Hieu Le, and Srijan Das. MS-Temba: Multi-Scale Temporal Mamba for Understanding Long Untrimmed Videos, 2025. arXiv:2501.06138. 1, 2
- [13] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2
- [14] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, pages 492–510, 2022. 2

A. Hyperparameters

Full hyperparameter list (optimizer, schedule, $\mathcal{L}_{\text{caus-cons}}$ weight, dropout, drop-path, weight decay) is reproduced from the training scripts shipped with the code release.

Hyperparameter	TSU		
	Attn.	Attn.X3	Causal
Padded temp. len.	2500	2500	2500
Optimizer	AdamW	AdamW	AdamW
Momentum	0.9	0.9	0.9
LR	4.5×10^{-4}	3.0×10^{-4}	2.0×10^{-4}
LR scheduler	cosine	cosine	cosine
Warmup epochs	10	12	10
Batch size	1	1	1
Weight decay	0.05	0.05	0.05
Dropout	0.25	0.0	0.15
Drop-path	0.1	0.1	0.15
Total epochs	250	250	200
$\mathcal{L}_{\text{caus-cons}}$ weight	–	–	1.0
$\mathcal{L}_{\text{caus-cons}}$ margin	–	–	0.1

Table 4. Training hyperparameters on TSU.

Hyperparameter	Charades		
	Attn.	Attn.X3	Causal
Padded temp. len.	256	256	256
Optimizer	AdamW	AdamW	AdamW
Momentum	0.9	0.9	0.9
LR	4.5×10^{-4}	3.0×10^{-4}	2.0×10^{-4}
LR scheduler	cosine	cosine	cosine
Warmup epochs	10	15	10
Batch size	5	5	5
Weight decay	0.05	0.07	0.05
Dropout	0.25	0.1	0.15
Drop-path	0.1	0.2	0.15
Total epochs	60	50	100
$\mathcal{L}_{\text{caus-cons}}$ weight	–	–	1.0
$\mathcal{L}_{\text{caus-cons}}$ margin	–	–	0.1

Table 5. Training hyperparameters on Charades.