

TBT-Former: Learning Temporal Boundary Distributions for Action Localization

Thisara Rathnayaka

Dept. of Computer Science & Engineering

University of Moratuwa

Sri Lanka

thisara.21@cse.mrt.ac.lk

Dr. Uthayasanker Thayasivam

Dept. of Computer Science & Engineering

University of Moratuwa

Sri Lanka

rtuthaya@cse.mrt.ac.lk

Abstract—Temporal Action Localization (TAL) remains a fundamental challenge in video understanding, aiming to identify the start time, end time, and category of all action instances within untrimmed videos. While recent single-stage, anchor-free models like ActionFormer have set a high standard by leveraging Transformers for temporal reasoning, they often struggle with two persistent issues: the precise localization of actions with ambiguous or “fuzzy” temporal boundaries and the effective fusion of multi-scale contextual information. In this paper, we introduce the Temporal Boundary Transformer (TBT-Former), a new architecture that directly addresses these limitations. TBT-Former enhances the strong ActionFormer baseline with three core contributions: (1) a higher-capacity scaled Transformer backbone with an increased number of attention heads and an expanded Multi-Layer Perceptron (MLP) dimension for more powerful temporal feature extraction; (2) a cross-scale feature pyramid network (FPN) that integrates a top-down pathway with lateral connections, enabling richer fusion of high-level semantics and low-level temporal details; and (3) a novel boundary distribution regression head. Inspired by the principles of Generalized Focal Loss (GFL), this new head recasts the challenging task of boundary regression as a more flexible probability distribution learning problem, allowing the model to explicitly represent and reason about boundary uncertainty. Within the paradigm of Transformer-based architectures, TBT-Former advances the formidable benchmark set by its predecessors, establishing a new level of performance on the highly competitive THUMOS14 and EPIC-Kitchens 100 datasets, while remaining competitive on the large-scale ActivityNet-1.3. Our code is available [here](#).

Index Terms—Temporal Action Localization, Transformers, Video Understanding, Boundary Regression, Feature Pyramid Network

I. INTRODUCTION

The ability to automatically understand human activities from video is a cornerstone of modern computer vision, with applications ranging from video summarization and surveillance to human-computer interaction [1]. A critical task in this domain is Temporal Action Localization (TAL), which requires not only classifying actions but also precisely identifying their temporal start and end points within long, untrimmed videos. The evolution of TAL methodologies has seen a significant shift from complex two-stage, proposal-based frameworks to more streamlined and efficient single-stage, anchor-free approaches [2], [3].

A seminal work in this new paradigm is ActionFormer [2], which demonstrated the remarkable efficacy of a minimalist,

Transformer-based architecture. By treating every moment in a video as a potential action center and using local self-attention within a multi-scale encoder, ActionFormer achieved state-of-the-art results without relying on pre-defined anchors or explicit proposal generation stages [2]. Despite its success, the ActionFormer framework and similar models exhibit two primary limitations that motivate our work.

First, the problem of **imprecise boundary regression** persists. Standard regression heads, typically optimized with an L1 (Least Absolute Deviations) or DIOU (Distance-Intersection over Union) loss, predict a single, deterministic offset for an action’s start and end times. This formulation is inherently inflexible and struggles when action boundaries are ambiguous—for instance, when an action begins or ends gradually. In such real-world scenarios, forcing the model to predict a single correct timestamp can lead to noisy training signals and inaccurate localization [4].

Second, the mechanism for **multi-scale feature fusion** can be improved. The feature pyramid in ActionFormer is constructed in a purely feed-forward manner, where strided operations generate progressively coarser feature maps [2]. This architecture lacks an explicit mechanism for coarser, semantically rich features to inform and refine the finer-grained features at higher temporal resolutions. This limits the model’s ability to integrate global context with precise local details, which is crucial for localizing actions of vastly different durations [2].

To address these challenges, we propose the Temporal Boundary Transformer (TBT-Former), a new model that introduces three targeted architectural advancements upon the ActionFormer baseline.

- 1) A **Scaled Transformer Backbone**: We increase the model’s representational capacity by scaling up the Transformer encoder, employing 16 attention heads and expanding the MLP hidden dimension by a factor of 6x. This allows the model to capture more intricate and long-range temporal dependencies.
- 2) A **Cross-Scale Feature Pyramid (CS-FPN)**: Inspired by refined Feature Pyramid Network (FPN) structures [5], we incorporate a top-down pathway with lateral connections. This architecture facilitates a bidirectional flow of information, enriching fine-grained temporal

features with high-level semantic context from coarser scales.

- 3) A novel **Boundary Distribution Regression (BDR) Head**: This is our main contribution. Drawing inspiration from the core philosophy of Generalized Focal Loss (GFL) [6], we reframe boundary prediction. Instead of regressing a single offset value, our BDR head learns a flexible probability distribution over a range of possible temporal locations. This allows the model to explicitly represent boundary uncertainty, leading to more robust and accurate localization, especially for actions with ambiguous transitions.

Through extensive experimentation, TBT-Former demonstrates superior performance, setting a new state-of-the-art on the THUMOS14 [7] and EPIC-Kitchens 100 [8] datasets and achieving highly competitive results on ActivityNet-1.3 [9]. Our work shows that explicitly modeling boundary uncertainty is a powerful and promising direction for advancing temporal action localization.

II. RELATED WORK

A. Single-Stage Temporal Action Localization

Early TAL methods often adopted a two-stage pipeline: first generating a sparse set of candidate temporal proposals, and then classifying and refining them [10], [11]. Models like BMN [10] achieved strong performance in proposal generation but were often computationally intensive and not end-to-end trainable. The field has since moved towards single-stage detectors, which perform localization and classification simultaneously in a single pass. A key innovation within this paradigm is the anchor-free approach, which discards pre-defined anchor windows and instead treats every temporal location on a feature map as a potential action candidate [2]. This design philosophy, exemplified by models like A2Net, GTAN, and notably ActionFormer, simplifies the architecture and has led to significant gains in both efficiency and accuracy [2]. TBT-Former builds directly upon this single-stage, anchor-free foundation, inheriting its efficiency while targeting its core representational limitations.

B. Multi-Scale Representations in Video

Actions in videos can span from a few seconds to several minutes, making multi-scale feature representation a critical component of any robust TAL model. The concept of a Feature Pyramid Network (FPN), originally developed for object detection, has been widely adopted to handle this variation [5]. An FPN creates a hierarchy of feature maps at different temporal resolutions, allowing the model to detect both short and long actions effectively. The baseline ActionFormer model implements a simple, feed-forward pyramid where deeper layers have coarser temporal resolution [2]. However, more advanced structures, such as the one explored in [5], employ top-down pathways and lateral connections to explicitly merge features across scales. This allows high-level semantic information from coarse feature maps to flow back and enrich the

spatially precise information in fine-grained maps, a principle we incorporate into our Cross-Scale FPN.

C. Boundary Modeling in Detection

A persistent challenge in localization tasks is the inherent ambiguity of object or action boundaries. Standard regression losses, such as L1 or Smooth L1, assume a single, deterministic ground-truth location, which is an oversimplification for many real-world cases. This issue was elegantly addressed in the object detection domain by Generalized Focal Loss (GFL) [6]. The core insight of GFL is to abandon the assumption of a Dirac delta distribution for a bounding box coordinate. Instead, it represents the coordinate as a flexible, arbitrary probability distribution learned over a set of discrete bins. The final coordinate is then computed as the expectation of this distribution, and a continuous version of Focal Loss is used for optimization [6]. This allows the model to represent its uncertainty; a sharp distribution indicates high confidence, while a flatter one signals ambiguity. In the TAL domain, TriDet [4] also explored modeling boundaries as distributions. Our proposed Boundary Distribution Regression (BDR) head is conceived as a direct and principled adaptation of the GFL philosophy to the temporal domain, providing a novel and effective solution for handling the pervasive issue of ambiguous action boundaries in videos.

III. OVERVIEW OF ACTIONFORMER

Our work builds upon the strong foundation of ActionFormer [2], a minimalist and highly effective single-stage, anchor-free framework for temporal action localization. The core philosophy of ActionFormer is to treat every moment in a video as a potential action candidate, thereby converting the complex task of structured output prediction into a more manageable sequence labeling problem. An overview of the ActionFormer architecture is provided in Figure 1.

The model follows a standard encoder-decoder structure. The encoder is a multi-scale Transformer network that processes a sequence of input video features [2]. It utilizes local self-attention to model temporal dependencies efficiently, even for long videos. A key design choice is the creation of a feature pyramid, where strided operations within the Transformer blocks generate feature maps at progressively coarser temporal resolutions. This multi-scale representation allows the model to effectively detect actions of varying durations [2].

The decoder consists of lightweight 1D convolutional networks that act as prediction heads. These heads are shared across all levels of the feature pyramid. For each temporal location on the pyramid, two parallel branches perform the main tasks: a classification head predicts the probability for each action class, and a regression head estimates the temporal distances from the current moment to the action’s start and end boundaries. The model is trained end-to-end using a combination of Focal Loss [12] for classification and a DIOU loss [13] for boundary regression [2]. This simple yet powerful design established a new state-of-the-art, providing a robust baseline for further innovation in the field.

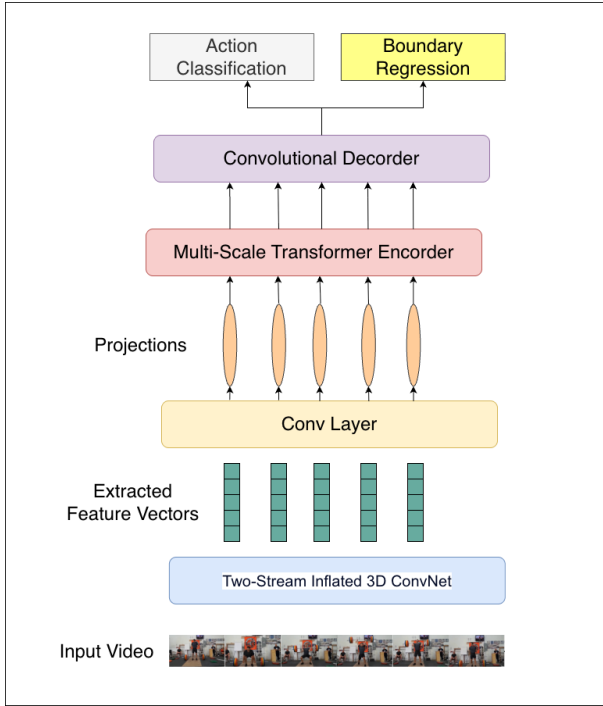


Fig. 1: An overview of the baseline ActionFormer architecture, illustrating the multi-scale Transformer encoder and the lightweight convolutional decoder with classification and regression heads [2].

IV. METHOD

A. Architecture Overview

TBT-Former builds upon the robust and efficient anchor-free architecture of ActionFormer [2]. The baseline model consists of a Transformer-based encoder that processes a sequence of video features, followed by a lightweight convolutional decoder with separate classification and regression heads. Our proposed TBT-Former retains this overall structure but introduces three significant enhancements: a scaled-up backbone for increased capacity, a cross-scale feature pyramid for improved multi-scale fusion, and a novel regression head for probabilistic boundary modeling. Figure 2 illustrates the overall architecture of TBT-Former, highlighting our key modifications.

B. Scaled Transformer Backbone

To enhance the model’s ability to learn complex temporal relationships and long-range dependencies, we first scale the capacity of the Transformer encoder. While the baseline ActionFormer uses a standard configuration for its Transformer blocks, we modify the key hyperparameters to create a more powerful backbone, as illustrated in Figure 3. Specifically, within each Multi-Head Self-Attention (MSA) block, we increase the number of parallel attention heads from the baseline’s 8 to 16. Furthermore, in the subsequent feed-forward network (FFN), we expand the hidden dimension by a factor of 6 relative to the input dimension, compared to the standard

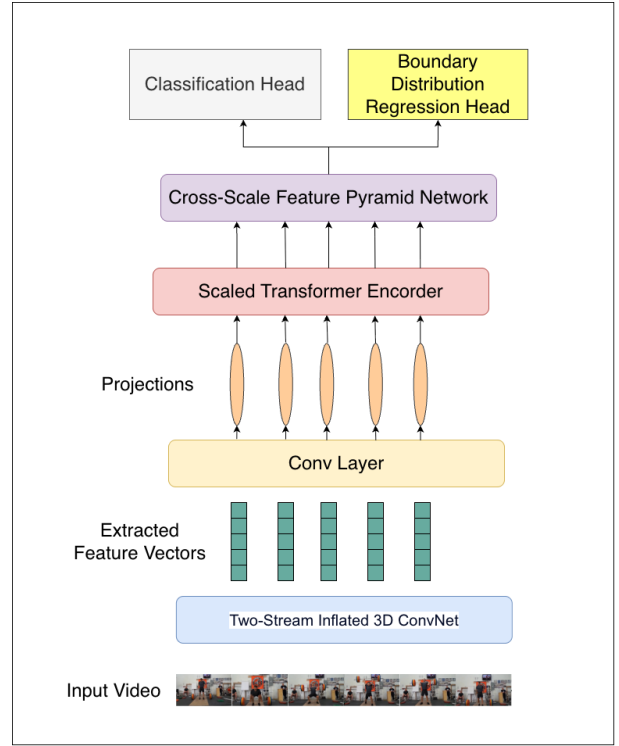


Fig. 2: Conceptual illustration of the TBT-Former architecture, highlighting the Scaled Backbone, Cross-Scale FPN (CS-FPN), and Boundary Distribution Regression (BDR) Head modules as key enhancements to the ActionFormer baseline.

4x expansion. This substantial increase in parameters provides the model with greater representational power, enabling it to capture more nuanced patterns in the temporal dynamics of actions, which is particularly beneficial for complex scenes and diverse datasets.

C. Cross-Scale Feature Pyramid (CS-FPN)

The original ActionFormer builds its feature pyramid $\{P_2, P_3, P_4, P_5\}$ in a simple feed-forward manner, where each level is generated by downsampling the previous one [2]. This design prevents the flow of information from semantically rich, coarse-grained levels back to temporally precise, fine-grained levels. To remedy this, we introduce a Cross-Scale Feature Pyramid (CS-FPN), inspired by the design in [5]. The overall architecture and the specific fusion operation are illustrated in Figure 4.

Our CS-FPN enhances the feature hierarchy with a top-down pathway and lateral connections. Let $\{C_2, C_3, C_4, C_5\}$ be the output feature maps from different stages of our scaled backbone, with decreasing temporal resolution. The pyramid construction proceeds as follows:

- 1) The coarsest feature map, C_5 , is first passed through a 1×1 convolutional layer to produce the initial pyramid map, P_5 .
- 2) For each subsequent level i (from 4 down to 2), the corresponding pyramid map P_i is computed by merging the

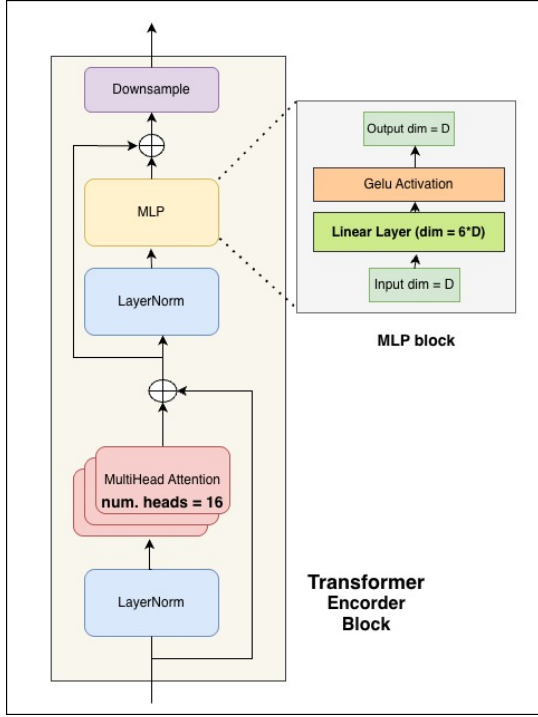


Fig. 3: The architecture of our Scaled Transformer Backbone unit. Compared to the baseline ActionFormer, we introduce two key enhancements for increased representational capacity: (1) the number of heads in the Multi-Head Attention module is increased to 16, and (2) the MLP block is expanded with a 6x hidden dimension, as shown in the zoomed-in view.

upsampled map from the coarser level, P_{i+1} , with the backbone map C_i . The merge operation is an element-wise addition:

$$P_i = \text{Conv}_{1 \times 1}(C_i) + \text{Upsample}(P_{i+1}) \quad (1)$$

where $\text{Conv}_{1 \times 1}$ is a lateral connection implemented as a 1D convolution to match channel dimensions, and Upsample is a temporal upsampling operator (e.g., nearest-neighbor interpolation).

- 3) Finally, each merged map P_i is processed by a 3×3 convolution to generate the final set of feature maps for the prediction heads. This refinement step helps to reduce aliasing artifacts from the upsampling and merge operations.

This architecture ensures that every level of the feature pyramid benefits from both high-resolution structural information (from the backbone) and high-level semantic context (from the top-down pathway), leading to a more powerful multi-scale representation for localizing actions of varying durations.

D. Boundary Distribution Regression (BDR) Head

Our most significant contribution is the Boundary Distribution Regression (BDR) head, which fundamentally changes how temporal boundaries are predicted.

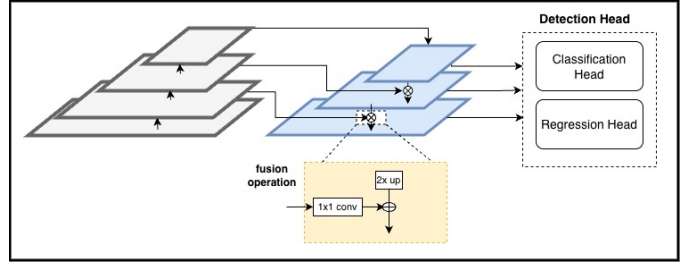


Fig. 4: The architecture of the Cross-Scale Feature Pyramid Network (CS-FPN). The fusion operation merges features from the bottom-up backbone pathway with the semantically richer features from the top-down pathway via lateral connections and element-wise addition.

Motivation: A standard regression head predicts a single float value for the start offset, d^s , and end offset, d^e . This formulation implicitly assumes that there is one precise, unambiguous boundary location. However, in reality, action boundaries are often fuzzy. Forcing the model to regress to a single point in such cases provides a noisy and unstable learning signal [4].

Probabilistic Formulation: Inspired by Generalized Focal Loss (GFL) [6], we reformulate this deterministic regression problem as a probabilistic one. Instead of predicting a single value d^s , our BDR head predicts a discrete probability distribution P_s over a pre-defined range of W possible integer offsets. The output for the start offset is a vector $P_s = \{p_s(0), p_s(1), \dots, p_s(W-1)\}$, where $\sum_{i=0}^{W-1} p_s(i) = 1$. The final continuous offset value, \hat{d}^s , is then calculated as the expectation of this distribution [6]:

$$\hat{d}^s = \sum_{i=0}^{W-1} i \cdot p_s(i) \quad (2)$$

An identical process is used to predict the end offset distribution P_e and compute \hat{d}^e . This formulation allows the model to express its uncertainty. A sharp, unimodal distribution for P_s indicates high confidence in a specific boundary, while a flatter or multimodal distribution reflects ambiguity, which is a more faithful representation of the underlying data [6].

Loss Function: To train the BDR head, we adapt the Distribution Focal Loss (DFL) from the GFL framework [6]. DFL provides an efficient and effective way to supervise the learning of this distribution. Given a continuous ground-truth offset d_{gt}^s , which lies between two discrete integer bins i and $i+1$ (i.e., $i \leq d_{gt}^s \leq i+1$), DFL focuses the learning signal exclusively on these two adjacent bins. The loss is calculated as a weighted cross-entropy [6]:

$$\mathcal{L}_{DFL}(P_s, d_{gt}^s) = -((i+1 - d_{gt}^s) \log(p_s(i)) + (d_{gt}^s - i) \log(p_s(i+1))) \quad (3)$$

This loss encourages the model to concentrate the probability mass of the distribution around the true continuous location. The total loss for our TBT-Former model is a combination of

the standard Focal Loss (\mathcal{L}_{cls}) for classification and our DFL for the start and end boundaries:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda(\mathcal{L}_{DFL}(P_s, d_{gt}^s) + \mathcal{L}_{DFL}(P_e, d_{gt}^e)) \quad (4)$$

where λ is a balancing hyperparameter. This loss structure provides a more stable and accurate learning objective for the challenging task of temporal boundary regression.

V. EXPERIMENTS

A. Datasets and Evaluation Metrics

We evaluate TBT-Former on three widely-used and challenging TAL benchmarks:

- **THUMOS14** [7]: This dataset is a standard benchmark for TAL, containing 20 action categories in over 400 untrimmed videos. It is characterized by long videos with a high density of action instances, making it particularly well-suited for evaluating localization precision.
- **ActivityNet-1.3** [9]: A large-scale dataset featuring 200 action classes in approximately 20,000 videos. Its scale and diversity make it an excellent testbed for model generalization and robustness.
- **EPIC-Kitchens 100** [8]: The largest dataset for egocentric action understanding. It contains numerous short, fine-grained actions defined by verb-noun pairs. Its unique characteristics test a model's ability to perform precise localization on brief, object-centric interactions.

For all datasets, we follow standard evaluation protocols and report the mean Average Precision (mAP) at various temporal Intersection over Union (tIoU) thresholds [2], [7].

B. Main Results

We compare TBT-Former with the ActionFormer baseline and other state-of-the-art methods.

Performance on THUMOS14: As shown in Table I, TBT-Former establishes a new state-of-the-art on the THUMOS14 benchmark. It achieves an average mAP of 68.0%, surpassing the strong ActionFormer baseline by 1.2 absolute points [2]. The inclusion of prior single-stage (GTAN, A2Net) and two-stage (BMN) methods further highlights the significant and consistent progress made, with TBT-Former outperforming all competitors across all tIoU thresholds [2]. This demonstrates the effectiveness of our architectural enhancements, particularly on a dataset that heavily rewards precise localization.

Performance on ActivityNet-1.3 and EPIC-Kitchens 100: Tables II and III summarize our results on the ActivityNet-1.3 and EPIC-Kitchens 100 validation sets, respectively.

On the large-scale ActivityNet-1.3 dataset, TBT-Former achieves highly competitive performance. As shown in Table II, our model is on par with the strong ActionFormer baseline, achieving an average mAP of 36.8%. While there is a minor dip at the tIoU=0.5 threshold, our model shows improved performance at the stricter thresholds of tIoU=0.75 and tIoU=0.95 [2]. This indicates superior localization precision for well-defined actions and demonstrates that our architectural changes maintain strong performance on large-scale benchmarks.

On EPIC-Kitchens 100, presented in Table III, TBT-Former consistently outperforms ActionFormer and prior methods for both Verb and Noun classification, achieving average mAP improvements of 1.0% and 1.2% over ActionFormer, respectively [2]. The substantial gain, especially for Noun classification, suggests that our BDR head is particularly effective for localizing the short, object-centric actions that are prevalent in egocentric video.

C. Ablation Studies

To validate our design choices and quantify the contribution of each new component, we conduct a systematic and extensive set of ablation studies on the THUMOS14 dataset.

1) *Summary of Component Contributions:* We first present a summary of how each of our proposed components contributes to the final performance. As shown in Table IV, each modification provides a clear and positive impact. The Scaled Backbone, CS-FPN, and BDR Head contribute +0.4, +0.3, and +0.8 mAP, respectively. When combined, they work synergistically to achieve a total improvement of +1.2 mAP over the strong ActionFormer baseline.

2) *Effect of Local Attention Window Size:* The original ActionFormer paper used a local attention window of size 19 for THUMOS14 [2]. Given that our Scaled Transformer Backbone has a higher representational capacity, we hypothesized that it could effectively leverage a larger temporal context. We experimented with several window sizes, with the results shown in Table V. Performance steadily improves as the window size increases from 19 to 30, where it peaks at an average mAP of 68.0%. Beyond this, performance begins to plateau, suggesting that a window size of 30 provides the optimal balance between contextual reach and model focus for our enhanced architecture.

3) *Design of the Feature Pyramid:* We validate the feature pyramid design inherited from ActionFormer [2]. As shown in Figure 5, performance is highly sensitive to the number of pyramid levels. A single-level model performs poorly (47.6% avg. mAP). Performance consistently increases with more levels, saturating at 6 levels, which confirms that a deep, multi-scale pyramid is critical for capturing actions of varying durations.

4) *Analysis of Loss Weight λ_{reg} :* We analyze the balance between the classification loss and our new DFL regression loss, controlled by the hyperparameter λ_{reg} . As shown in Table VI, the model is robust to a range of values, but performance peaks at $\lambda_{reg} = 1.0$. This confirms that an equal weighting between the classification and boundary distribution regression objectives provides the optimal trade-off for our model.

5) *Alternative Backbone Architectures:* Finally, we explored replacing the Transformer-based backbone of the baseline model with two alternative architectures to understand their suitability for TAL. The results are summarized in Table VII.

Hybrid Transformer-Mamba Backbone: We designed a hybrid encoder that uses Transformer blocks for high-resolution feature maps and Mamba blocks for coarser, low-

TABLE I: Comparison with state-of-the-art methods on the THUMOS14 test set. TBT-Former achieves the highest performance across all metrics.

Model	Type	Avg. mAP	mAP@0.3	mAP@0.4	mAP@0.5	mAP@0.6	mAP@0.7
GTAN [2]	Single-Stage	57.8	70.4	63.3	57.8	47.2	33.5
A2Net [2]	Single-Stage	58.6	70.6	65.0	59.9	51.3	37.5
BMN [2]	Two-Stage	56.0	67.5	60.9	56.0	47.4	34.9
ActionFormer [2]	Single-Stage	66.8	82.1	77.8	71.0	59.4	43.9
TBT-Former (Ours)	Single-Stage	68.0	82.5	79.0	72.4	60.6	45.3

TABLE II: Performance on ActivityNet-1.3 validation set. The average mAP is calculated over [0.5:0.05:0.95]. *Results for ActionFormer are from the original paper [2].

Model	mAP@0.5	mAP@0.75	mAP@0.95	Avg. mAP
PGCN [3]	48.3	33.2	3.3	31.1
BMN [10]	50.1	34.8	8.3	33.9
G-TAD [11]	50.4	34.6	9.0	34.1
AFSD [15]	52.4	35.3	6.5	34.4
ActionFormer [2]	54.7	37.8	8.4	36.6*
TBT-Former (Ours)	53.9	38.2	8.5	36.8

TABLE III: Performance on EPIC-Kitchens 100 validation set for Verb and Noun tasks. The average mAP is over [0.1:0.1:0.5].

Task	Model	mAP@0.1	mAP@0.2	mAP@0.3	mAP@0.4	mAP@0.5	Avg. mAP
Verb	BMN [10]	10.8	9.8	8.4	7.1	5.6	8.4
	G-TAD [11]	12.1	11.0	9.4	8.1	6.5	9.4
	ActionFormer [2]	26.6	25.4	24.2	22.3	19.1	23.5
	TBT-Former (Ours)	27.2	26.7	25.6	22.9	20.1	24.5
Noun	BMN [10]	10.3	8.3	6.2	4.5	3.4	6.5
	G-TAD [11]	11.0	10.0	8.6	7.0	5.4	8.4
	ActionFormer [2]	25.2	24.1	22.7	20.5	17.0	21.9
	TBT-Former (Ours)	26.4	25.9	23.8	21.7	17.9	23.1

resolution maps. Mamba is a recent State Space Model (SSM) architecture with linear-time complexity, making it efficient for long sequences [14]. The core of Mamba is the Selective State Space Model (SSSM), which maps an input $x(t)$ to an output

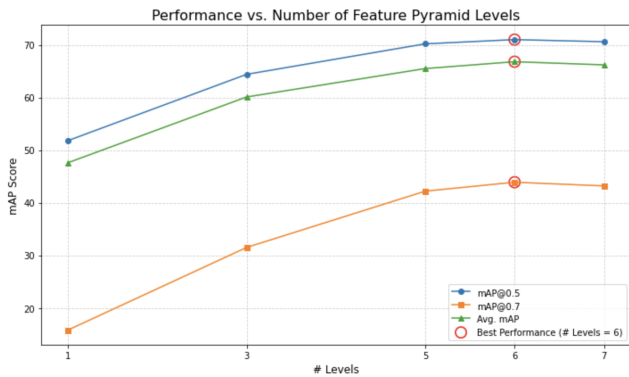


Fig. 5: Ablation on the number of feature pyramid levels. Performance across different metrics (mAP@0.5, mAP@0.7, and Avg. mAP) peaks at 6 levels, indicating this is the optimal depth for the pyramid in our architecture.

TABLE IV: Summary of component-wise contributions on THUMOS14.

#	Model Configuration	Avg. mAP	Δ
1	Baseline (ActionFormer)	66.8	-
2	+ Scaled Backbone	67.2	+0.4
3	+ Cross-Scale FPN (CS-FPN)	67.1	+0.3
4	+ Boundary Distribution Head (BDR)	67.6	+0.8
5	Full Model (TBT-Former)	68.0	+1.2

TABLE V: Ablation on the local attention window size. All experiments use the full TBT-Former model.

Window Size	mAP@0.5	mAP@0.7	Avg. mAP
19 (Baseline size)	71.5	44.2	67.1
25	72.0	44.8	67.6
30	72.4	45.3	68.0
37	72.2	45.1	67.8

$y(t)$ via a latent state $h(t)$. Its discretized form is a recurrence:

$$\begin{aligned} h_k &= \bar{A}h_{k-1} + \bar{B}x_k \\ y_k &= Ch_k \end{aligned} \quad (5)$$

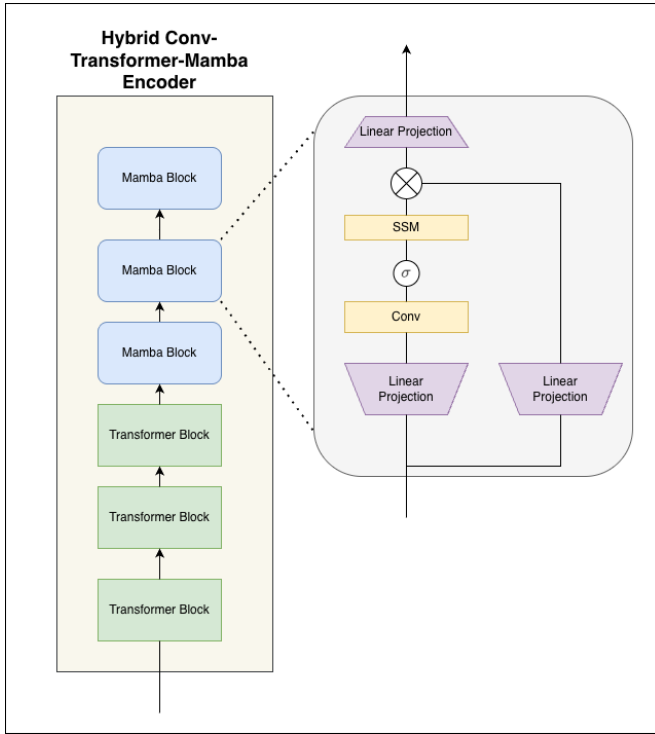
Mamba’s innovation is making the parameters \bar{B} and \bar{C} input-dependent, allowing it to selectively focus on parts of the sequence [14]. Our hybrid architecture, shown in Figure 6(a), achieved a competitive 65.8% avg. mAP.

SGP-based Backbone: We also implemented a fully-convolutional backbone using the Scalable-Granularity Perception (SGP) layer from [4]. The SGP layer, shown in Figure 6(b), replaces self-attention with two parallel branches: an instant-level branch to enhance feature discriminability and a window-level branch with parallel convolutions to capture multi-granularity context. The SGP operation is formulated as [4]:

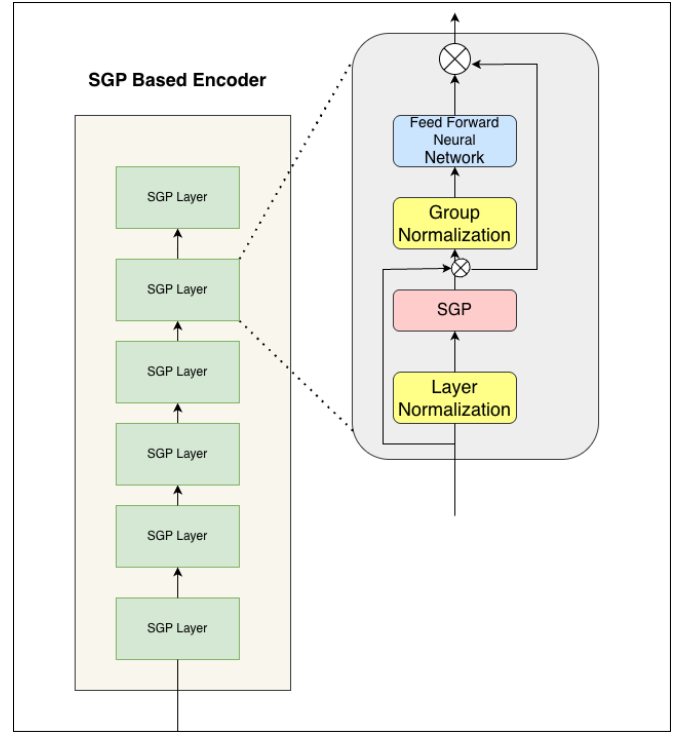
$$f_{SGP} = \phi(x)FC(x) + \psi(x)(Conv_w(x) + Conv_{kw}(x)) + x \quad (6)$$

where $\phi(x)$ and $\psi(x)$ are gating functions, and FC denotes a fully-connected layer. This model achieved 66.2% avg. mAP.

While both alternatives show strong performance, they do not surpass the optimized Transformer baseline. This suggests that the local self-attention mechanism remains highly effective for temporal feature extraction in TAL, though the competitive results of SGP and Mamba highlight them as promising directions for future research.



(a) Hybrid Conv-Transformer-Mamba encoder.



(b) SGP-based encoder.

Fig. 6: Architectures of the alternative backbones explored in our ablation study. (a) The hybrid approach begins with convolutional embedding and an SE block, followed by a stem of Transformer blocks. The branch creates a feature pyramid using Transformer blocks at fine-grained levels and Mamba blocks at coarse-grained levels. (b) The SGP-based encoder is built from SGP blocks, with a two-branch design for feature discrimination and multi-granularity context.

TABLE VI: Ablation on the regression loss weight λ_{reg} .

λ_{reg}	mAP@0.5	mAP@0.7	Avg. mAP
0.2	70.1	39.8	65.0
0.5	71.4	41.7	66.4
1.0	71.0	43.6	66.9
2.0	69.7	43.1	66.3
5.0	68.8	42.5	65.1

TABLE VII: Ablation on alternative backbone architectures, compared to the ActionFormer baseline.

Backbone Type	mAP@0.5	mAP@0.7	Avg. mAP
Transformer (Baseline)	71.0	43.9	66.8
Hybrid Attn-Mamba	69.5	42.9	65.8
SGP (Convolutional)	70.1	43.2	66.2

VI. CONCLUSION

In this paper, we introduced TBT-Former, a novel single-stage, anchor-free model for temporal action localization. Our work makes three key contributions: a higher-capacity scaled Transformer backbone for enhanced temporal reasoning, a cross-scale feature pyramid for improved multi-scale fusion, and a novel Boundary Distribution Regression (BDR) head for modeling temporal ambiguity.

Within the paradigm of Transformer-based architectures,

ActionFormer established a formidable performance benchmark. Our results demonstrate that TBT-Former successfully advances this benchmark, setting a new level of performance on the highly competitive THUMOS14 and EPIC-Kitchens 100 datasets, while remaining competitive on the large-scale ActivityNet-1.3. Our most significant finding is that explicitly modeling temporal boundaries as probability distributions is a powerful and promising paradigm.

Furthermore, our ablation studies on alternative backbones, such as the convolutional SGP and the SSM-based Mamba architectures, revealed their potential as efficient and competitive alternatives to self-attention. While they did not surpass our optimized Transformer-based model in this work, their strong performance indicates a valuable direction for future research. Subsequent work could focus on refining these architectures or developing more sophisticated hybrid models, potentially unlocking further gains in both performance and computational efficiency.

REFERENCES

- [1] B. Wang, Y. Zhao, L. Yang, T. Long, and X. Li, "Temporal Action Localization in the Deep Learning Era: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 4, pp. 2171-2190, 2023.
- [2] C.-L. Zhang, J. Wu, and Y. Li, "ActionFormer: Localizing Moments of Actions with Transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 492-510.

- [3] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph Convolutional Networks for Temporal Action Localization," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2019.
- [4] D. Shi et al., "TriDet: Temporal Action Detection with Relative Boundary Modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023.
- [5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 2117-2125.
- [6] X. Li et al., "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," in *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2020.
- [7] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The THUMOS challenge on action recognition for videos 'in the wild'," *Computer Vision and Image Understanding*, vol. 155, pp. 1-23, 2017.
- [8] D. Damen et al., "Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100," *Int. J. Comput. Vis.*, vol. 130, no. 1, pp. 33-55, 2022.
- [9] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 961-970.
- [10] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "BMN: Boundary-Matching Network for Temporal Action Proposal Generation," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2019.
- [11] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-TAD: Sub-Graph Localization for Temporal Action Detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.
- [12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980-2988.
- [13] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," in *Proc. AAAI Conf. on Artificial Intelligence*, 2020.
- [14] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," *arXiv preprint arXiv:2312.00752*, 2023.
- [15] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Learning Salient Boundary Feature for Anchor-free Temporal Action Localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.
- [16] F. Cheng and G. Bertasius, "Tallformer: Temporal action localization with a long-memory transformer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 503-521.
- [17] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," *arXiv preprint arXiv:2103.03404*, 2021.
- [18] J. Kim, M. Lee, and J.-P. Heo, "Self-feedback detr for temporal action detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 10 286-10 296.
- [19] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213-229.
- [20] T. Tran, T.-D. Truong, and Q.-H. Bui, "Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization," *arXiv preprint arXiv:2303.09055*, 2023.
- [21] S. Lee, J. Jung, C. Oh, and S. Yun, "Enhancing temporal action localization: Advanced s6 modeling with recurrent mechanism," *arXiv preprint arXiv:2407.13078*, 2024.