TRACKNETV5: ROBUST SHUTTLECOCK TRACKING VIA MOTION PROMPTS AND SPATIOTEMPORAL ATTENTIVE FUSION

Run-Lin Chang, Yu-Shuen Wang, Jiun-Long Huang

National Yang Ming Chiao Tung University, Hsinchu City, Taiwan

ABSTRACT

Shuttlecock tracking is challenging due to the object's small size, extreme speed, and severe motion blur. While current models like TrackNetV4 achieve high precision, they rely on 2D convolutions to implicitly handle temporal relationships, failing to capture complex spatiotemporal dynamics. We propose TrackNetV5, which introduces multi-scale $R(2+1)D$ blocks for explicit spatiotemporal feature extraction and a gated residual motion prompt layer for high-quality motion attention. Furthermore, a spatial-channel-gate module adaptively fuses motion priors with visual features across multiple decoder layers. Experimental results show that TrackNetV5 achieves a state-of-the-art F1-Score of 98.47. We also present TrackNetV5-Lite, which maintains a high F1-Score of 98.06 with only 2.58M parameters, demonstrating significant potential for real-time edge deployment.

***Index Terms*—** Shuttlecock tracking, Badminton, R(2+1)D, Motion prompts, Attention mechanism, Edge computing

1. INTRODUCTION

In modern sports analytics, accurate ball trajectory data is essential for tactical analysis and athlete training [1, 2, 3]. However, shuttlecock tracking in badminton remains a formidable challenge due to its diminutive size, extreme velocities, and the resulting severe motion blur and frequent occlusions.

The TrackNet series [4, 5] has established the benchmark for this task, evolving from heatmap-based predictions to sophisticated architectures incorporating trajectory rectification [6] and motion attention [7]. Despite these advancements, a fundamental architectural limitation persists: these models rely heavily on 2D CNN-based backbones. This approach processes temporal relationships *implicitly* through channel-wise frame stacking, which fails to capture complex spatiotemporal dynamics effectively. Even recent modular enhancements like TrackNetV4 [7], which introduce motion prompts, still operate on a static 2D backbone and limit feature fusion to high-level stages, resulting in a lack of deep, multi-scale spatiotemporal reasoning.

To address these gaps, we propose TrackNetV5, a robust spatiotemporal framework that shifts from implicit to explicit modeling. Our approach introduces multi-scale $R(2+1)D$ blocks [8] to decompose 3D convolutions into spatial and temporal components, enabling deeper spatiotemporal feature extraction. To further enhance robustness, we utilize a gated residual motion prompt layer [9] that generates high-quality motion attention maps while suppressing background noise and flicker. These motion priors are adaptively integrated with visual features across multiple decoder levels via a newly proposed spatial-channel-gate (SCG) module. By leveraging an asymmetric encoder-decoder design with EfficientNetV2 blocks [10], TrackNetV5 effectively balances computational efficiency with high-precision localization.

Experimental results demonstrate that TrackNetV5 sets a new state-of-the-art (SOTA) with an F1-Score of 98.47. Notably, our lightweight variant, TrackNetV5-Lite, achieves 98.06 F1-Score with only 2.58M parameters, demonstrating its superior efficiency and readiness for edge deployment in real-time sports broadcasting and training systems.

2. RELATED WORK

2.1. Object Detection Frameworks

Object detection has evolved from region-proposal methods like faster R-CNN [11] to real-time single-stage detectors such as the YOLO series [12, 13, 14]. These frameworks typically operate on a frame-by-frame basis, prioritizing the localization of relatively large objects with moderate movement. While recent advancements like YOLOv12 [15] introduce attention-centric architectures (e.g., area attention) to expand receptive fields, they remain ill-suited for badminton tracking. The aggressive downsampling in these backbones often leads to the loss of fine-grained features for minuscule objects. Furthermore, the lack of temporal context makes it nearly impossible for static detectors to distinguish a blurred shuttlecock (i.e., traveling at speeds up to 426 km/h) from complex background noise.

2.2. Visual Tracking and Siamese Networks

Visual tracking methodologies, notably siamese networks like SiamFC [16], utilize cross-correlation between a target template and a search region. A fundamental assumption in these models [16, 17, 18, 19] is that the object's displacement

between frames is small enough to stay within a predefined search radius. In professional sports, however, the shuttlecock's velocity frequently exceeds these spatial constraints, leading to tracking drift. Moreover, without explicit temporal modeling, these methods struggle to re-acquire the target after occlusion or when the background contains visually similar artifacts.

2.3. Evolution of the TrackNet Series

The TrackNet family was specifically engineered for high-speed sports through heatmap-based prediction, evolving from basic encoder-decoder structures [4, 5] to models with trajectory rectification [6] and motion attention [7]. While these versions improved precision, they primarily rely on channel-wise frame stacking for *implicit* temporal modeling. Even the motion prompt layer in TrackNetV4 [7] functions as a plug-and-play module on a static 2D backbone, limiting its ability to reason over deep spatiotemporal dynamics. In contrast, our proposed TrackNetV5 shifts toward an explicit spatiotemporal architecture. By integrating $R(2+1)D$ factorization and multi-level gated fusion, our method captures the shuttlecock's motion cues more robustly across different scales than the previous generation of stacked-frame models.

3. METHODOLOGY

3.1. Network Architecture Overview

The proposed TrackNetV5 is an end-to-end spatiotemporal U-Net architecture designed for robust shuttlecock tracking. The process begins with an input tensor of shape (T, C, H, W), where the parameters T, C, H, W denote the number of frames, number of channels, height, and width, respectively. The output is subsequently branched into two paths:

1. **Motion Branch:** The *motion prompt layer* generates a motion prior, which is subsequently downsampled via *MaxPool* layers to match the spatial resolutions of various decoder stages.

2. **Visual Branch:** The features pass through an *R(2+1)D* block for spatiotemporal factorization and are reshaped to $(T \cdot C, H, W)$ for the main backbone.

The architectural design of TrackNetV5 follows an asymmetric encoder-decoder design [20] to balance high-precision localization with computational efficiency. As illustrated in Fig. 1, the encoder leverages Fused-MBConv blocks (i.e., introduced by EfficientNetV2 [10]) to prioritize rapid feature extraction in the early stages. These features are then passed via skip connections to the decoder, which employs standard MBConv blocks integrated with the newly developed SCG module for adaptive spatiotemporal fusion.

To ensure the architecture remains versatile across different hardware constraints, we design TrackNetV5 as a scalable model family rather than a fixed configuration. Inspired by EfficientNetV2 [10] and MobileNetV2 [21], we introduce a width multiplier α to systematically adjust the network's capacity. For any layer with an initial channel count C, the scaled number of filters C_α is defined as:

$$C_\alpha = \alpha \times C \tag{1}$$

By modulating α, we can assign additional computational resources to the network width to capture finer spatial patterns or reduce them for efficiency. While the standard version ($\alpha = 1.0$) is optimized for high-fidelity tracking, the *TrackNetV5-Lite* variant is derived by applying a reduced multiplier (e.g., $\alpha = 0.5$). This systematic scaling allows the lite version to maintain a compact parameter count of only 2.58M, ensuring robust performance and real-time inference on power-constrained edge devices.

3.2. Spatiotemporal Modeling with R(2+1)D Blocks

Previous TrackNet methods primarily rely on 2D convolutions, where consecutive frames are stacked along the channel dimension. This approach treats temporal sequences as static spatial representations, failing to explicitly capture the frame-by-frame evolution of high-speed trajectories. To address this limitation, we introduce R(2+1)D convolutional blocks to factorize spatiotemporal learning into decoupled spatial and temporal components.

While a standard 3D convolutional kernel of size $t \times d \times d$ (where t denotes the temporal extent and d the spatial size) is the most direct approach to capturing complex temporal dynamics, it entails a significant increase in parameter count and computational complexity. To mitigate this, we decompose the 3D kernel into a spatial 2D convolution of size $1 \times d \times d$, followed by a temporal 1D convolution of size $t \times 1 \times 1$. This structural factorization offers significant advantages in both optimization and efficiency. By decoupling spatial appearance from temporal dynamics, the $R(2+1)D$ block simplifies the learning process while maintaining high representational power. This design effectively reduces the computational burden and parameter count to strike an optimal balance between spatiotemporal modeling and inference speed, fulfilling the requirements for real-time edge deployment.

3.3. Gated Residual Motion Prompt Layer

To capture the complex dynamics of fast-moving shuttlecocks, we propose an enhanced gated residual motion prompt layer. While TrackNetV4 [7] utilizes static frame differencing, our module employs a *gated* strategy designed to simultaneously suppress background noise and preserve fine-grained motion details.

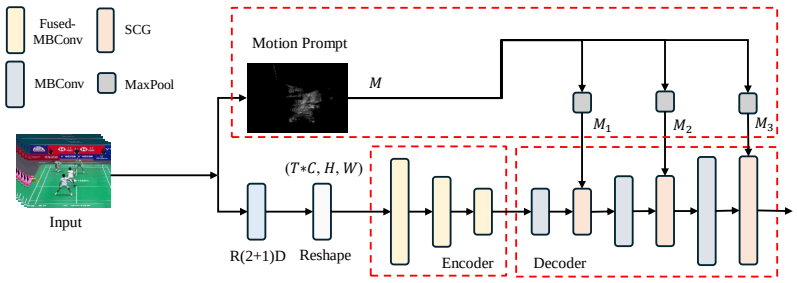

Fig. 1. The architecture of TrackNetV5. The network explicitly models spatiotemporal dynamics through R(2+1)D blocks. Motion priors are integrated into the MBConv-based decoder via multi-level SCG fusion.

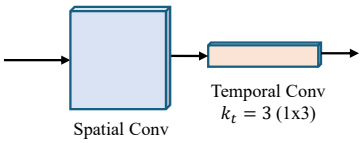

Fig. 2. R(2+1)D block architecture. The 3D convolution is decomposed into a spatial convolution and a temporal convolution to capture both appearance and motion dynamics.

Initial motion extraction. As illustrated in Fig. 3, the input tensor is a grayscale sequence $\mathcal{X}$ that focuses on intensity fluctuations. We compute absolute frame differences to highlight pixel-wise motion, which are then temporally aggregated into a mean difference map:

$$D = \frac{1}{T-1} \sum_{t=1}^{T-1} |\mathcal{X}_{t+1} - \mathcal{X}_t|. \tag{2}$$

Here, $D \in \mathbb{R}^{1\times H\times W}$ represents the raw motion intensity. This initial step effectively isolates the moving shuttlecock from the stationary court background.

Gated residual refinement. The core innovation lies in the gated residual mechanism, which refines the raw signal D through two parallel paths:

- **Gate Path** (g): This path generates spatial weights to mitigate artifacts like camera flicker. The map D is smoothed via average pooling and a convolutional block $\mathcal{G}$, followed by a sigmoid activation modulated by a learnable temperature τ:

$$g = \sigma\left(\frac{\mathcal{G}(\text{AvgPool}(D))}{\tau}\right) \tag{3}$$

  The parameter τ (constrained within $[0.25, 4.0]$) regulates gating confidence; a higher τ flattens the activation curve to ensure stable gradient flow during early training.

- **Transformation Path** ($\mathcal{T}$): Simultaneously, D passes through a depthwise convolutional bottleneck $\mathcal{T}$. This path recalibrates motion magnitudes and preserves sharp spatial details that may be lost during the smoothing process in the Gate Path.

Motion attention generation. The refined motion feature F is then formulated via a gated residual connection:

$$F = (1 - g) \cdot D + g \cdot \mathcal{T}(D) \tag{4}$$

Finally, the motion attention map M is produced by a linear transformation followed by a sigmoid activation:

$$M = \sigma(\alpha \cdot F + \beta) \tag{5}$$

where α and β are learnable parameters. This architecture allows the network to dynamically retain original motion cues while adaptively integrating noise-suppressed features, providing a high-quality spatiotemporal prior for subsequent fusion stages.

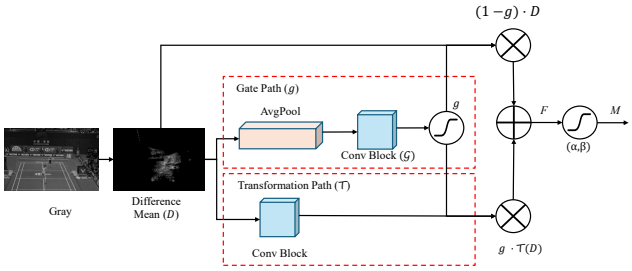

Fig. 3. Architecture of the gated residual motion prompt layer. The module extracts grayscale frame differences and utilizes a gated mechanism to adaptively fuse raw motion signals with transformed features, regulated by a learnable temperature τ.

3.4. Multi-Level Fusion via Spatial-Channel-Gate

The core innovation of the TrackNetV5 decoder is the SCG module. Inspired by the convolutional block attention module [22], SCG is designed to adaptively recalibrate feature responses by integrating the motion prior M across multiple

scales (M_1, M_2, M_3) in Fig. 1. Unlike the simple single-stage fusion in previous works [7], SCG employs parallel spatial and channel attention paths as shown in Fig. 4.

Dual-gate mechanisms. The SCG module processes visual features V by identifying both the relevance and location of motion through two gates:

- **Channel gate** ($g_c \in \mathbb{R}^{C\times1\times1}$): This path identifies "what" motion features are relevant by aggregating the global context from both visual features V and the prior scaled motion M. We apply global average pooling to both inputs, followed by a multi-layer perceptron with a bottleneck design to compute the channel-wise attention weight:

$$g_c = \sigma(\text{MLP}([\text{GAP}(V), \text{GAP}(M)])) \tag{6}$$

- **Spatial gate** ($g_s \in \mathbb{R}^{1\times H\times W}$): This path focuses on "where" the shuttlecock is likely located. It computes a spatial attention map by applying a convolutional layer directly to the motion prior:

$$g_s = \sigma(\text{Conv}_{k\times k}(M)) \tag{7}$$

Feature integration and residual learning. The visual features V are first modulated by these two gates via element-wise multiplication ($\otimes$). To ensure feature integrity and stable gradient flow, this gated representation is processed through a lightweight mixing block $\mathcal{N}$ (consisting of depthwise-separable convolutions) before being combined with the original features via a residual connection:

$$V_{out} = V + \mathcal{N}((g_s \otimes V) \otimes g_c) \tag{8}$$

This design allows the model to selectively emphasize spatiotemporal regions of interest while maintaining the stability of the underlying visual representations.

3.5. Loss Function and Deep Supervision

The total objective function combines a primary tracking loss and a deep supervision component [23]:

$$\mathcal{L}_{total} = \mathcal{L}_{wbce} + \mathcal{L}_{ds}, \tag{9}$$

where $\mathcal{L}_{wbce}$ is the weighted binary cross-entropy that is used to address the extreme class imbalance between the shuttlecock and the background. The second term, $\mathcal{L}_{ds}$, is a deep supervision strategy [23] that injects auxiliary objective functions into the hidden layers to mitigate vanishing gradients and encourage the learning of discriminative features in earlier stages of the decoder. By forcing intermediate layers to produce coarse heatmaps that approximate the ground truth, the network achieves faster convergence and improved feature representation.

3.6. Scalable Architecture via Width Multiplier

To facilitate real-time deployment across diverse hardware constraints, we introduce TrackNetV5-Lite. Following the design philosophy of MobileNet [21], we employ a width multiplier α as a hyperparameter to construct a scalable model family. The design is governed by two key principles:

- **Uniform Channel Thinning**: We explicitly introduce the width multiplier α to customize the network's capacity according to the target computational budget. For a given layer with C input channels, the lightweight variant constructs a uniformly thinned representation with $C' = \alpha \times C$. By setting $\alpha = 0.5$, we effectively reduce the computational cost and memory footprint quadratically. This mechanism allows TrackNetV5 to be tailored for resource-constrained edge devices, providing a flexible trade-off between inference latency and tracking accuracy.
- **Efficient Spatiotemporal Factorization**: TrackNetV5-Lite inherits the R(2+1)D block strategy from the main architecture, which decomposes complex volumetric interactions into spatial and temporal components. This design was explicitly chosen to ensure scalability; it allows TrackNetV5-Lite to preserve deep spatiotemporal reasoning capabilities without the cubic computational complexity of standard 3D kernels. Consequently, the model retains explicit motion modeling essential for accurate tracking, even under strict computational constraints.

3.7. Implementation Details

The model is optimized using the AdamW optimizer with an initial learning rate of 1×10^{-4}, a weight decay of 5×10^{-4}, and a batch size of 16. We employ a cosine decay to regulate the learning rate, with specific decay milestones established at the 100 and 200 epochs to facilitate stable convergence. All experiments were conducted using a fixed random seed to ensure reproducibility.

4. EXPERIMENTS

4.1. Datasets and Implementation Details

We evaluate TrackNetV5 on the public badminton dataset [5, 6] with frames resized to 512×288. The value of T (e.g., sequence length) is set to 5. A prediction is considered a True Positive (TP) if the distance to the ground truth is within 4 pixels [5].

4.2. Quantitative Results and Comparison

Table 1 compares TrackNetV5 with SOTA baselines. TrackNetV5 achieves a leading F1-Score of 98.47, surpassing

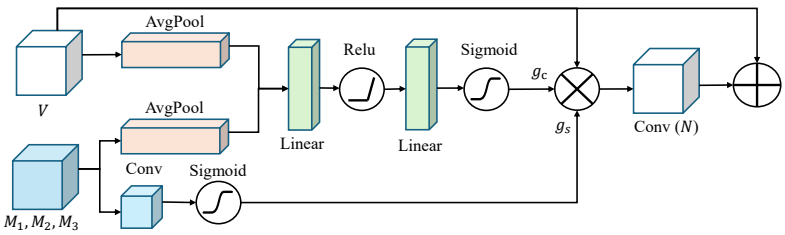

Fig. 4. Architecture of the proposed SCG. The module utilizes parallel spatial and channel attention mechanisms inspired by CBAM [22] to fuse visual features V with motion priors M, incorporating a residual connection for robust feature preservation.

Table 1. Performance comparison on the shuttlecock dataset.

Method	Params	Acc.	Prec.	Recall	F1	FPS
TrackNetV3 [6]	11.34M	95.98	99.67	95.69	97.64	119.09
TrackNetV4 [7]	11.34M	96.58	99.56	96.48	97.99	118.81
TrackNetV5	10.09M	**97.37**	**99.72**	**97.25**	**98.47**	60.7
TrackNetV5-Lite	**2.58M**	96.69	99.59	96.57	98.06	**135.18**

Table 2. Effect of loss function.

Loss function	Acc.	Prec.	Recall	F1
$\mathcal{L}_{wbce}$	96.42	99.53	96.33	97.90
$\mathcal{L}_{wbce} + \mathcal{L}_{ds}$	**96.50**	**99.56**	**96.39**	**97.95**

TrackNetV3 and TrackNetV4. Notably, the lightweight TrackNetV5-Lite (2.58M parameters) reaches an F1-Score of 98.06 while running at 135.18 FPS on an RTX 5080, effectively doubling the accuracy gain over the baseline with superior speed.

Table 2 shows the effect of loss function. Experimental results show that adding $\mathcal{L}_{ds}$ into loss function can improve accuracy, precision, recall and F1, supporting its benefit for optimization and decoder feature learning. Table 3 shows the effect of α. Experimental results show a clear trade-off: larger α improves accurcy, precision, recall and F1 but increases model size and reduces speed, while smaller α yields a lighter and faster model. This confirms α as an effective knob for adapting TrackNetV5 to different deployment constraints.

Table 3. Effect of α.

α	Params	Acc.	Prec.	Recall	F1
1	10.09M	**97.37**	**99.72**	**97.25**	**98.47**
0.75	6.46M	97.07	99.64	96.97	98.29
0.5	**2.58M**	96.69	99.59	96.57	98.06

4.3. Ablation Study

The architecture of TrackNetV5 is developed by systematically refining and extending the TrackNetV4 framework.

Table 4. Ablation study.

Method	Acc.	Prec.	Recall	F1
TrackNetV4 [7]	96.58	99.56	96.48	97.99
+ MBConv	96.73	99.61	96.62	98.09
+ Motion Prompt + SCG	**96.77**	**99.63**	**96.64**	**98.11**
+ R(2+1)D	96.64	99.56	96.55	98.03

To analyze the contribution of each proposed component, we conduct an ablation study by progressively upgrading the baseline TrackNetV4 architecture. The results of this incremental integration are summarized in Table 4. The architectural evolution begins by replacing standard convolutions with MBConv blocks. This modification significantly enhances the model's recall. Building upon this, we integrate the motion prompts coupled with the SCG module. This combination yields a clear improvement in tracking performance, as the SCG module effectively fuses visual features and motion priors. Finally, the addition of R(2+1)D blocks completes the framework, providing the necessary spatiotemporal factorization to capture complex shuttlecock dynamics.

5. CONCLUSION

In this paper, we present TrackNetV5, a robust spatiotemporal framework for high-speed shuttlecock tracking. By introducing explicit spatiotemporal extraction via R(2+1)D blocks mechanism, the model successfully disentangles complex motion dynamics from static background noise. The integration of an optimized gated residual motion prompt layer and the multi-level spatial-channel-gate fusion module ensures precise localization even under severe motion blur and occlusion. Experimental results demonstrate that TrackNetV5 achieved the highest detection accuracy compared to the baselines. Furthermore, the TrackNetV5-Lite variant offers a highly efficient solution for edge deployment, achieving superior performance with only 2.58M parameters. Future work will focus on extending this architecture to other fast-paced ball sports and integrating multi-object tracking to enable comprehensive, automated match analysis systems.

6. REFERENCES

[1] Denys Rozumnyi, Jan Kotera, Filip Sroubek, Lukas Novotny, and Jiri Matas, "The world of fast moving objects," in *CVPR*, 2017, pp. 5203–5211.

[2] Tianxiao Zhang, Xiaohan Zhang, Yiju Yang, Zongbo Wang, and Guanghui Wang, "Efficienst golf ball detection and tracking based on convolutional neural networks and kalman filter," *arXiv preprint arXiv:2012.09393*, 2020.

[3] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, no. 1, pp. 7068349, 2018.

[4] Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsì-Uí İk, and Wen-Chih Peng, "Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications," in *AVSS*, 2019, pp. 1–8.

[5] Nien-En Sun, Yu-Ching Lin, Shao-Ping Chuang, Tzu-Han Hsu, Dung-Ru Yu, Ho-Yi Chung, and Tsì-Uí İk, "Tracknetv2: Efficient shuttlecock tracking network," in *ICPAI*, 2020, pp. 86–91.

[6] Yu-Jou Chen and Yu-Shuen Wang, "Tracknetv3: Enhancing shuttlecock tracking with augmentations and trajectory rectification," in *ACM Multimedia Asia*, 2023, pp. 1–7.

[7] Arjun Raj, Lei Wang, and Tom Gedeon, "Tracknetv4: Enhancing fast sports object tracking with motion attention maps," in *ICASSP*, 2025, pp. 1–5.

[8] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *CVPR*, 2018, pp. 6450–6459.

[9] Qixiang Chen, Lei Wang, Piotr Koniusz, and Tom Gedeon, "Motion meets attention: Video motion prompts," in *ACML*, 2024.

[10] Mingxing Tan and Quoc Le, "Efficientnetv2: Smaller models and faster training," in *ICML*, 2021, pp. 10096–10106.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[14] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *CVPR*, 2023, pp. 7464–7475.

[15] Yunjie Tian, Qixiang Ye, and David Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[16] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*, 2016, pp. 850–865.

[17] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *CVPR*, 2019, pp. 4282–4291.

[18] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*, 2017, pp. 1763–1771.

[19] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank, "Learning attentions: residual attentional siamese network for high performance online visual tracking," in *CVPR*, 2018, pp. 4854–4863.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.

[21] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[22] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "Cbam: Convolutional block attention module," in *ECCV*, 2018, pp. 3–19.

[23] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu, "Deeply-supervised nets," in *AISTATS*, 2015, pp. 562–570.