Towards Highly Effective Moving Tiny Ball Tracking via Vision Transformer

Jizhe Yu[1(✉)], Yu Liu[1], Hongkui Wei[2], Kaiping Xu[1], Yifei Cao[1], and Jiangquan Li[1]

[1] DUT School of Software Technology and DUT-RU International School of Information Science and Engineering, Dalian University of Technology, Dalian, China
yujizhe@mail.dlut.deu.cn

[2] State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, China

Abstract. Recent tiny ball tracking methods based on deep neural networks have significantly progressed. However, since moving balls in the video are always blurred, most existing methods cannot achieve accurate tracking due to limited receptive fields and sampling depth. Furthermore, as high-resolution competition videos become increasingly common, existing methods perform poorly on high-resolution images. To this end, we provide a strong baseline for tracking tiny balls called TrackFormer. Firstly, we use Vision Transformer to build the whole network architecture and enhance the tiny ball localization through its powerful spatial mining ability. Secondly, we develop a Global Context Sampling Module (GCSM) to capture more powerful global features, thereby increasing the accuracy of tiny ball identification. Finally, we design a Context Enhancement Module (CEM) to enhance tiny ball semantics to achieve robust tracking performance. To promote research and development of tiny ball tracking, we establish a Large-scale Tiny Ball Tracking dataset called LaTBT. Specifically, LaTBT is founded on three types of tiny balls (badminton, tennis, and squash), offering more than 300 video sequences and over 223K annotations from 19 types of professional matches to address various tracking challenges in diverse and complex backgrounds. To our knowledge, LaTBT is the first large-scale dataset for tiny ball tracking. Experiments demonstrate that our baseline achieves state-of-the-art performance on our proposed benchmark dataset. The dataset and the algorithm code are available at https://github.com/Gi-gigi/TrackFormer.

Keywords: Visual tracking · Tiny ball tracking · Benchmark dataset · Transformer baseline · Global context guidance

1 Introduction

Tiny ball tracking is a crucial research field in precision sports science [1]. The trajectory, location, and usage information of tiny balls enhance the spectatorship of matches and aid in match analysis and judgment, thus attracting significant research interest. Although existing tiny ball tracking algorithms have made tremendous progress, the increasing resolution of television broadcasts today has far exceeded their performance capabilities.

D.-S. Huang et al. (Eds.): ICIC 2024, LNCS 14864, pp. 368–379, 2024.
https://doi.org/10.1007/978-981-97-5588-2_31

As shown in Fig. 1(a), TrackNet [2, 3], based on the VGG backbone, can not provide a satisfactory viewing experience compared to Ground Truth (GT). This is mainly because the backbone network was initially designed for image classification and is unsuitable for training tiny ball datasets. Furthermore, processing high-resolution images leads to a significant increase in memory usage. Therefore, developing a cost-effective and high-performing tiny ball tracking network is crucial for match analysis.

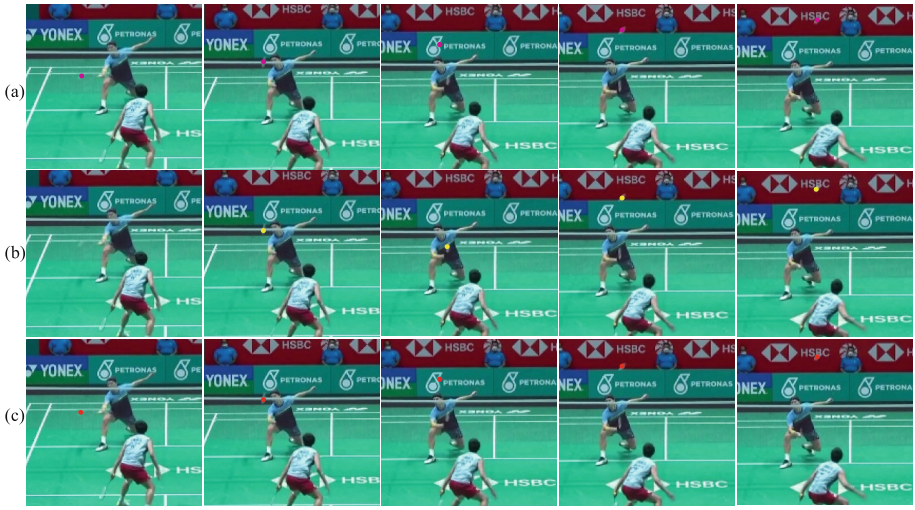

Fig. 1. Visual comparison with state-of-the-art method. (a) Input image and GT(purple circle). (b) TrackNet [3] (yellow circle). (c) Our method (red circle). (Color figure online)

To this end, we propose a robust baseline method called TrackFormer, which adopts Vision Transformer as an architecture to overcome the limitations of fixed receptive field and limited sampling depth of traditional tracking networks. It is also able to capture semantic information crucial for tracking balls. Furthermore, many Transformer-based trackers have demonstrated excellent performance on multiple tracking datasets [4, 5], which can accurately locate and identify objects in the scene due to their excellent spatial mining capabilities. For this purpose, our TrackFormer can precisely track the ball frame-by-frame, as shown in Fig. 1(b). Our method can still locate the ball efficiently even in highly blurry conditions (1st and 3rd frames). To enhance the robustness of tracking performance, this paper proposes a Global Context Sampling Module (GCSM) to obtain global context features and further promote the ball localization ability of the network. To effectively address the issue of low image quality caused by high-speed moving tiny balls, we also design a Context Enhancement Module (CEM) to minimize background noise interference, enabling the network to effectively utilize global context for guiding top-down and achieve better tracking performance.

We believe another significant reason for blocking tracking capability is the drawbacks of existing tracking datasets. In the field of tiny ball tracking, the existing TrackNet tracking benchmark consists of two widely used datasets. However, both datasets are of small scale, not only limited in the variety of sports but also lacking in the diversity of

competition environments, rendering them unable to represent the challenges encountered in tracking tiny balls in professional competitions. Taking TrackNet's badminton dataset as an example, it includes videos from only a single amateur or professional match, limiting the applicability of tracking methods and performing poorly in challenging scenarios. To provide a comprehensive evaluation platform for tiny ball tracking, we contribute a large-scale tracking benchmark called LaTBT. Firstly, this dataset is a pioneering large-scale, high-resolution (1080 p) tiny ball tracking dataset, containing over 300 video clips and more than 223 K manually annotated binary heatmaps with frames and resolutions far exceeding that of popular tiny ball tracking datasets. Secondly, we meticulously annotate each frame to ensure coverage of a broader range of tiny ball tracking challenges, such as blur, afterimage, overlap, and other low-quality image problems often missing in existing datasets. Moreover, LaTBT contains tiny ball data from 19 types of professional competitions, which is suitable for a wide range of professional game scenarios and can effectively evaluate the performance of different tracking algorithms in actual scenarios. Thus, our LaTBT dataset is distinguished not only by its high diversity but also by its significant challenges. Our contributions are as follows:

- In this paper, we propose a novel transformer-based baseline method for tiny ball tracking called TrackFormer, which consists of two main components, namely the Global Context Sampling module (GCSM) and the Context Enhancement module (CEM). With the cooperation of the two modules, compared with the traditional tracking method, the proposed method effectively balances the relationship between sampling depth and receptive field, thereby improving the tracking performance.
- We are the first to propose a large-scale dataset for tiny ball tracking. LaTBT includes vast data, rich scene diversity, and careful annotations. We analyze the properties, advantages, and uniqueness of our benchmarks and compare them with other datasets in Sect. 3.
- Benchmark tests on the proposed dataset demonstrate that our baseline method achieves unprecedented performance in tiny ball tracking. We also report the results of six additional state-of-the-art trackers on LaTBT to be more convincing. Moreover, we run at the speed of 30 FPS on a single GPU.

2 Related Work

Recently, many trackers have been proposed to address the various challenges in tiny object tracking. Yu et al. [6] combine object detection frameworks with deep neural networks to track small objects in airborne images. Zhu et al. [7] utilize knowledge distillation networks to enhance the recognition capability of tiny objects. Yang et al. [8] propose a relational inference network to track small objects in satellite videos through semantic relations among frames. Wu et al. [9] adopt unclassified backbone networks to enhance infrared small object detection performance. Sun et al. [3] develop a lightweight tiny ball tracking network called TrackNet to achieve real-time tracking of shuttlecocks in videos. In this paper, we primarily focus on tracking tiny balls.

Outstanding tracking performance relies on a robust network architecture. Inspired by UNet [18], TrackNet [2, 3] employs a symmetrical encoder-decoder architecture to

capture coarse-to-fine image features, demonstrating exceptional potential in tracking tiny balls. As followers of UNet, Qin et al. [10] adopt a two-level nested U-shaped structure to acquire multi-scale features and detect small salient objects without changing resolution. To further enhance model performance, Wu et al. [9] introduce a multi-scale aggregation module to improve the tracking of infrared small objects. Additionally, recent successes in applying Transformers from language processing to video tracking tasks [4, 5] have shown that transformer-based networks can effectively model the global context [20], thus enhancing the localization and identification of the ball. In our work, we draw inspiration from these methods, including U-shaped networks, Transformers, and multi-scale aggregation, but uniquely incorporate these approaches in our model.

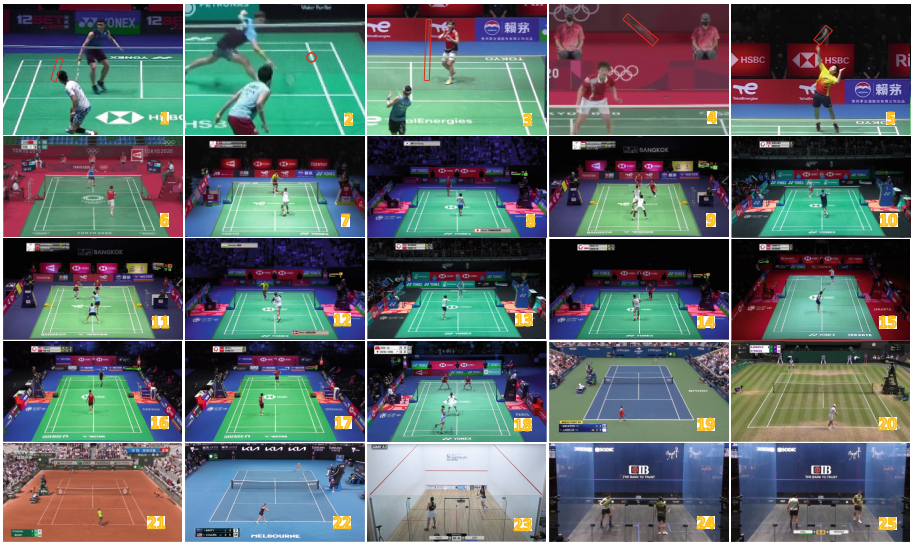

Fig. 2. Sample from our LaTBT.

3 Our Tiny Ball Tracking Benchmark Dataset

3.1 TrackNet Tracking Benchmark

The current tiny ball tracking datasets face three significant challenges. Firstly, they lack the necessary data scale to represent the complex and diverse environment of actual games. Most of these datasets are derived from amateur competition videos, and the size of the ball is fixed on a regular scale, which does not account for changes in rotation, scaling, or deformation. Secondly, the datasets consist of low-resolution, low-frame-rate images (720p 25fps), which do not align with current criteria for image processing application. Additionally, there are concerns about the quality of annotations, with issues such as extreme blur (Fig. 1), overlap (Fig. 2), and scale variation (Fig. 3) frequently omitted from existing datasets.

3.2 LaBTB Tracking Benchmark

To solve the above problems, we create a Large-scale Tiny Ball Tracking dataset called LaTBT, which consists of three types of tiny ball sports: badminton, tennis, and squash. First, regarding dataset scale and professionalism. The badminton dataset is derived from live broadcasts of globally renowned badminton tournaments in 2022 (Winter Olympics Badminton, World Badminton Championships, Thomas Cup, Uber Cup, Sudirman Cup, All England Open, Malaysia Open, Indonesia Open, Denmark Open, Swiss Open, Thailand Open, German Open, and French Open), covering men's singles, women's singles, men's doubles, and women's doubles, with more than 220 video clips and 141,151 frames annotated from approximately 80 h of video. The tennis dataset is selected from Grand Slam Tennis in 2022, including men's and women's singles, with 50,732 annotated frames and more than 70 video clips. The squash dataset is from the semifinals and finals of the World Championship in 2022, with 31,427 frames annotated and 30 video clips. All the full videos can be publicly accessed through YouTube at https://www.youtube.com/@bwftv. Secondly, we provide intuitive visual examples, as shown in Fig. 1, 2, 3, 4 and 5 show examples of different ball sizes, No. 6–18 show the complex and diverse badminton dataset, No. 19–22 show the tennis dataset, and No. 23–25 show the squash dataset. Furthermore, we create a low-resolution (720p 25fps) badminton test dataset containing 28,220 frames, selected from the 2018 Women's Singles World Championships and the All England Finals, to evaluate the model's generalization ability. Then, all three datasets choose high-resolution (1080p) videos, with frame rates including 25fps, 30fps, and 60fps. Lastly, more than three subjects carefully checked our tiny ball dataset to avoid low-level annotation errors. In summary, the LaTBT dataset surpasses existing ball tracking datasets in scale, professionalism, resolution, diversity, and challenges. To our knowledge, our benchmark is the largest, highly diverse, highly professional, high-resolution, and high-quality annotated tiny ball tracking dataset.

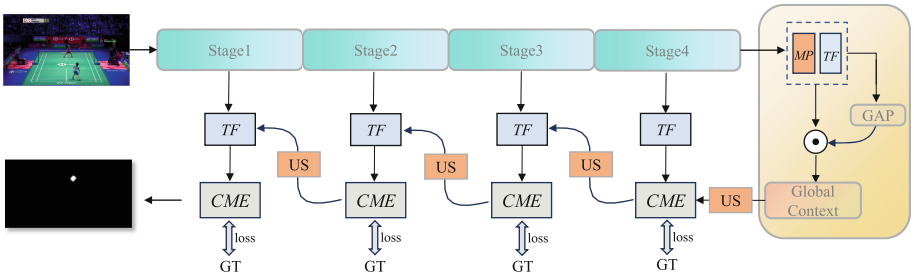

Fig. 3. The framework of our proposed tiny ball tracking network TrackFormer with two key modules: Global Context Sampling Module and Context Enhancement Module, which cooperated to improve tiny ball tracking performance.

4 Method

4.1 Overview Architecture

We adopt PVT-v2 as the encoder backbone, whose Transformer architecture offers excellent global perceptual ability and efficient computational performance. To better capture global context information, we downsample the network and design a Global Context Sampling Module (GCSM) to model deep features. Subsequently, we develop a Context Enhancement Module (CEM) that effectively integrates global context with decoder features to reduce background noise interference [18], enhancing ball semantics and improving tracking performance. The network architecture is shown in Fig. 3.

4.2 Global Context Sampling Module

In the above, we have mentioned that global context information plays a crucial role in enhancing the tracking performance of the tiny ball. Therefore, we design the GCSM to extract deep semantic information by learning the global view of the entire image, as shown in Fig. 3. Next, we clarify the design details of the GCSM.

Suppose that the feature output from the top layer of the encoder is X. To increase the sampling depth, we downsample the input feature map by a factor of 2 using max-pooling. Subsequently, a transformer block $\mathcal{TF}$ and $\mathcal{MLP}$ from the original Transformer [20] are used for global context modeling. This process can be described as

$$X_g = \mathcal{MLP}(\mathcal{TF}(Reshape(Mp(X)))) \tag{1}$$

where Mp is the max-pooling downsampling. $Reshape$ represents the tensor change from $[B, C, H, W]$ to $[B, H \times W, C]$. Moreover, Transformer $\mathcal{TF}$ can be described as

$$\mathcal{TF} = Softmax\left(QK^T \Big/ \sqrt{d}\right)V \tag{2}$$

To obtain a global view of the entire image, we first employ Global Average Pooling (GAP) to acquire a global attention map X_a. Then, X_a is used to adaptively weight X_g to enhance the location information of the foreground object:

$$X_a = \sigma\left(GAP\left(X_g\right)\right) \tag{3}$$

$$X_g^{\omega} = X_g \odot X_a \tag{4}$$

where σ is the Sigmoid nonlinear activation function. $\odot$ represents element-wise multiplication and X_g^{ω} is denoted as the after weighted X_g.

In summary, our GPSM can enlarge the receptive field and sampling depth to localize the tiny ball while delivering global context to the Context Enhancement Module (CEM) to mitigate the issues of ball semantics dilution and inaccurate localization.

4.3 Context Enhancement Module

Intuitively, global context features and initial features from the encoder are two types of inconsistent features. If they are directly integrated, it may lead to feature misalignment and introduce a lot of background noise [14, 18], which is detrimental to tiny ball tracking (such as "TrackNet" from Fig. 1). For this reason, we construct CEM with two purposes: to enhance the semantics of the ball guided by global context and to effectively integrate two complementary features to mitigate feature differences.

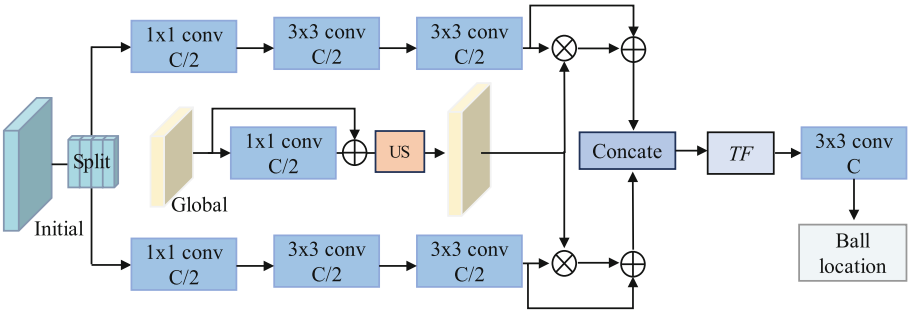

Fig. 4. Illustration of Context Enhancement Module (CEM) for effective feature fusion.

As shown in Fig. 4. Suppose that the initial feature is $F_i^S \in \mathcal{R}^{N\times C\times H\times W}(S = 1 \sim 4$, C, H, W are the encoder stages, channel number, height, and width). To fully utilize the global context to guide feature fusion, we first adopt two 1×1 convolution layers to obtain branch features, namely $B_1 \in \mathcal{R}^{N\times C/2\times H\times W}$ and $B_2 \in \mathcal{R}^{N\times C/2\times H\times W}$. After that, two layers of 3×3 convolution followed by BN and Relu layers are passed, respectively. The above process can be described (taking B_1 as an example) as follows:

$$B_1 = conv_2^{3\times3}(conv_1^{3\times3}(conv_1^{1\times1}(F_i^S))) \tag{5}$$

Since both the global context and the initial feature have different resolutions, we first interpolate the global context feature $G \in \mathcal{R}^{N\times C\times H\times W}$ to match the dimensions of the feature map B_1, and then is processed by the residual block to generate strengthened residual context:

$$G_{res} = conv^{1\times1}(Up(G)) + G \tag{6}$$

Next, we aggregate the processed initial features and the global context. First G_{res} is evenly split into two feature maps $G_{res}^1 \in \mathcal{R}^{N\times C/2\times H\times W}$ and $G_{res}^2 \in \mathcal{R}^{N\times C/2\times H\times W}$ along the channel dimension. We then apply the global contexts G_{res}^1 and G_{res}^2 to the features B_1 and B_2 respectively, and adopt B_1 and B_2 to obtain the identity mappings, which can be formulated as (taking B_1 branch as an example):

$$G_{res}^1 = \mathcal{F}_{split}(G_{res}) \tag{7}$$

$$M_1 = G_{res}^1 \odot B_1 + B_1 \tag{8}$$

where $\mathcal{F}_{split}(\cdot)$ denotes the split operation along channel dimension. $M_1 \in \mathcal{R}^{N\times C/2\times H\times W}$ is feature attention map after merged.

Finally, we concatenate the two branch features and further refine them with the original transformer $\mathcal{TF}$, and then a 3×3 convolutional layer is used to determine the final location of the ball:

$$P = \sigma\left(conv^{3\times 3}(\mathcal{TF}(M_1 \oplus M_2))\right) \tag{9}$$

Where $\oplus$ represents the concatenation. In conclusion, the proposed CEM can correct feature misalignment and achieve self-refinement under the guidance of global context to improve ball tracking performance.

For training, we apply the BCE loss as used in U^{2}Net [10] for each decoder stage:

$$\mathcal{L}_{bce} = -\sum_{x=1}^{H}\sum_{y=1}^{W}\left[G(x,y)log(P(x,y)) + (1-G(x,y))log(1-P(x,y))\right] \tag{10}$$

where $G(x, y)$ and $P(x, y)$ are the ground truth label and the predicted label at the location (x, y), respectively. H and W are the height and width of the images, respectively.

Therefore, the total loss of our training is defined as follows:

$$\mathcal{L} = \sum_{s=1}^{4}\mathcal{L}_{bce}^{s}(P(x,y), G(x,y)) \tag{11}$$

where the lowercase subscript s represents each decoder stage. During inference, we integrate the features of the various stages, like U^{2}Net, to generate the final prediction result.

5 Experiments

5.1 Experiments Setup

Evaluation Datasets. We mainly use our proposed benchmark datasets to evaluate the performance of our method along with other state-of-the-art methods. We adhere to the dataset partition ratios in [11] to divide the tiny ball tracking datasets into training, validation, and test sets. Specifically, LaTBT comprises 147,385 images for training, 29,030 images for validation, and 46,895 images for testing. Since badminton is the fastest ball sport, we also evaluate the model's generalization ability on our proposed low-resolution badminton dataset (LR Test).

Implementation Details. Our proposed TrackFormer and all the comparative algorithms are trained from scratch in an end-to-end manner. We run all experiments on the publicly available Pytorch 1.5.0 platform. An RTX 2080Ti GPU card (with 12 GB memory) is used for training and testing. During network training, each frame is first preprocessed to [960 × 960] and then resize to [224 × 224] for the PVT [12] backbones, and data augmentation methods such as normalizing and flipping are used. Our encoder parameters are initialized from PVT. Adam optimizer [13] with default hyperparameters is adopted to train the network. We train the network for 50 epochs with a batch size of 16. The initial learning rate is lr = 1e−5, and warm-up and linear decay strategies adjust the learning rate. During testing, each image is resized to [1024 × 1024] and then passed into the network without any post-processing.

Evaluation Metrics. In this study, the confusion matrix is chosen as the primary evaluation metric [3] to evaluate the performance of the model. The sigmoid function converts the network output to score maps between 0 and 1, which are then classified into pixels values of 0 or 1 using a threshold of 0.5. The predicted ball location is determined as the center of the largest area in the heatmap. A tolerance value (tol) is set to measure the network's success in identifying the ball, with tol being determined as 10 based on the average diameter of all tiny balls appearing in the frames.

When there is no ball detected by the model, the output will show TN, which means that there is no ball within the frame. On the other hand, if the model identifies a non-ball object, the output will show FP. However, when a ball is visible within the frame, the output will show TP if the model correctly identifies the ball. This means that the difference between the predicted position by the model and the ground truth position is less than the defined tolerance value (tol). In cases where the model fails to identify the ball, the output will show FN1. Meanwhile, if the model identifies an object but the predicted distance is greater than the tolerance value (tol) from the ground truth position, the output will show FN2. Accuracy, Precision, and Recall can be formulated as follows:

$$Accuarcy = {}^{(TP+TN)}\!/_{(TP+TN+FP+FN1+FN2)} \tag{12}$$

$$Precision = {}^{TP}\!/_{TP+FP} \tag{13}$$

$$Recall = {}^{TP}\!/_{TP+FN1+FN2} \tag{14}$$

To balance the relationship between precision and recall, $F_\beta(\beta{=}1)$ score [19] is used as a comprehensive indicator, and $F_\beta(\beta{=}2)$ emphasizes the importance of recall.

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{15}$$

5.2 Comparison with State-of-the-Arts

We compare the proposed TrackFormer with six recent state-of-the-art models, including Vgg-TrackNet and Res-TrackNet [2][3], UIUNet [9], InSPyReNet [14], RDIAN [16], AGPCNet [15], and MTU-Net [17]. For a fair comparison, the predicted maps are either provided by the authors or generated by the officially released pre-trained models. It is worth noting that the comparison algorithms we selected are mainly from infrared tiny object detection and salient object detection, primarily because they are highly similar to our task, and both can track tiny objects. During the training, we employ multiple frames input and single frame outputs (MISO) design strategy [3] to achieve simple ball tracking.

Quantitative Evaluation. Table 1 reports the quantitative results of the LaTBT benchmark and LR Test, in which we compare our method with the six state-of-the-art algorithms in terms of Accuracy, F_1, and F_2. Our TrackFormer tracking neytwork achieves

the best results across all metrics on the LaTBT benchmark. Notably, our method prioritizes high recall to minimize tracking omissions, as missing a positive instance is more severe than mistakenly identifying a false positive. Consequently, our method achieves the highest F_2 score. Other comprehensive indicators Acc and F_1 also demonstrate the excellent tracking ability of our method and show superior performance compared to the VIT-based MTU-net. We also test the model trained on LaTBT on our proposed large LR Test, and the results consistently show our method's outstanding performance. This indicates its strong ability to handle challenging inputs. Figure 5 shows the accuracy-measure curve, further demonstrating our tracking method's superior performance. At different tolerance levels, the red curve of our method is significantly higher than the other curves on both datasets, underscoring the effectiveness of our proposed method in tracking tiny balls. Apart from the visualizations in Fig. 1, we also provide clear and easy-to-understand visualization videos from https://github.com/Gi-gigi/TrackFormer

Table 1. Comparison of TrackFormer with state-of-the-art Tracking methods. The best performance in each column is highlighted in bold.

Summay	Speed	Param	LaTBT			LR Test		
Method	FPS	M	Acc	F_1	F_2	Acc	F_1	F_2
Vgg-TrackNet	36.2	26.45	.864	.919	.882	.737	.809	.733
Res-TrackNet	14.05	35.41	.909	.950	.930	.765	.836	.776
InSPyReNet	12	28.10	.934	.963	.953	.790	.855	.797
UIUNet	28.4	50.54	.948	.971	.964	.814	.873	.821
RDIAN	32.4	17.08	.925	.956	.937	.809	.871	.828
AGPCNet	20.4	52.51	.951	.973	.974	.817	.878	.838
MTU-Net	25.01	57.03	.954	.974	.968	.821	.882	.847
Ours	31.7	47.06	**.963**	**.979**	**.976**	**.854**	**.905**	**.879**

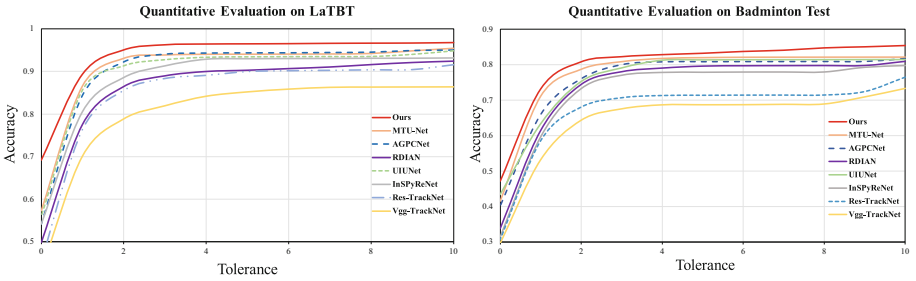

Fig. 5. Accuracy-measure curves of the proposed method and other SOTA algorithms on two benchmark datasets.

Additionally, we list the parameters and speeds of each method to measure efficiency. Our method is more lightweight and achieves real-time speed compared to the latest SOTA methods, with highly competitive performance. Another method, RDIAN, shows considerable performance gaps although it has a faster inference speed. In summary, TrackFormer achieves a balance between efficiency and performance.

Table 2. Ablation study with different components combinations on LaTBT.

No	Components Setting	LaTBT		
		Acc	F_1	F_2
1	Baseline	.907	.946	.928
2	+GCSM	.950	.972	.962
3	+ GCSM + CEM	.963	.979	.976

5.3 Ablation Study

We conduct the ablation study on each key component designed in this paper, using PVT-v2 [12] as the backbone on the LaTBT dataset. As shown in Table 2, the model integrating all components (GCSM and CEM) achieved the best performance, highlighting their importance in achieving superior tracking results. The "Baseline" model (No. 1) adopts a structure similar to UNet [18]. Introducing the GCSM module (No. 2) significantly increases accuracy from 0.907 to 0.950. Furthermore, adding the CEM module (No. 3) further increases the accuracy by 6% compared to the Baseline model (No. 1). The F_1 and F_2 scores also show gradual increases. In summary, the two components complement each other and jointly enhance the robustness of tiny ball tracking.

6 Conclusion

We propose a novel tiny ball tracking baseline, called TrackFormer, to identify and locate tiny balls from low quality video frames. This network optimizes tracking performance through design two novel modules: the Global Context Sampling Module (GCSM) and the Context Enhancement Module (CEM). GCSM improves ball location by deepening sampling depth, while CEM mitigates background noise to enhance ball semantics. By integrating these modules, TrackFormer achieves highly effective tiny ball tracking. To provide a comprehensive evaluation platform, we introduce a large-scale, challenging, and diverse tracking benchmark, called LaTBT. Experimental results show that TrackFormer effectively addresses challenges such as blur, afterimage, and overlap in low-quality images on LaTBT. Future work will collect a broader range of tiny ball tracking datasets, including table tennis, baseball, and ice hockey. et al., to develop a more comprehensive evaluation platform. We also plan to borrow ideas from single-stage object detectors to enhance performance further.

Acknowledgments. This work is Supported by the National Natural Science Foundation of China under Grant 61672128, and in part by the Dalian Key Field Innovation Team Support Plan under Grant 2020RT07.

References

1. Chu, W.-T., Situmeang, S.I.G.: Badminton video analysis based on spatiotemporal and stroke features. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (2017)
2. Huang, Y., et al.: TrackNet: a deep learning network for tracking high-speed and tiny objects in sports applications. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 1–8 (2019)
3. Sun, Nien-En et al. TrackNetV2: efficient shuttlecock tracking network. In: 2020 International Conference on Pervasive Artificial Intelligence, pp. 86–91 (2020)
4. Paul, M., et al.: Robust visual tracking by segmentation. arXiv preprint arXiv:2203.11191 (2022)
5. Cao, S., et al.: SwinCGH-Net: enhancing robustness of object detection in autonomous driving with weather noise via attention. In: International Conference on Intelligent Computing (2023)
6. Yu, M., Leung, H.: Small-object detection for UAV-based images. In: 2023 IEEE International Systems Conference, pp. 1–6 (2023)
7. Zhu, Y., et al.: Tiny object tracking: a large-scale dataset and a baseline. IEEE Trans. Neural Networks and Learning Systems PP (2022)
8. Yang, X., et al.: Relation learning reasoning meets tiny object tracking in satellite videos. IEEE Trans. Geosci. Remote Sens. (2024)
9. Wu, X., et al.: UIU-Net: U-Net in U-Net for infrared small object detection. IEEE Trans. Image Process. **32**, 364–376 (2022)
10. Qin, X., et al.: U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection. Pattern Recognit. **106**, 107404 (2020)
11. Tian, X., et al.: Bi-directional object-context prioritization learning for saliency ranking. In; 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5872–5881 (2022)
12. Wang, W., et al.: PVT v2: improved baselines with pyramid vision transformer. Comput. Visual Media **8**, 415–424 (2021)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR. arXiv preprint arXiv:1412.6980 (2014)
14. Kim, T., et al.: Revisiting image pyramid structure for high resolution salient object detection. In: Asian Conference on Computer Vision (2022)
15. Zhang, T., et al.: AGPCNet: attention-guided pyramid context networks for infrared small target detection. arXiv preprint arXiv:2111.03580 (2021)
16. Sun, H., et al.: Receptive-field and direction induced attention network for infrared dim small target detection with a large-scale dataset IRDST. IEEE Trans. Geosci. Remote Sens. **61**, 1–13 (2023)
17. Wu, T., et al.: MTU-Net: Multilevel TransUNet for Space-Based Infrared Tiny Ship Detection. IEEE Trans. Geosci. Remote Sens. **61**, 1–15 (2022)
18. Liu, J., et al.: PoolNet+: exploring the potential of pooling for salient object detection. IEEE Trans. Pattern Anal. Mach. Intell. **45**, 887–904 (2022)
19. Hamed, B.A., Ibrahim, O.A.S., Abd, E.-H.: Optimizing classification efficiency with machine learning techniques for pattern matching. J. Big Data **10**(1), 124 (2023)
20. Liu, N., et al.: Visual saliency transformer. In: 2021 IEEE/CVF International Conference on Computer Vision, pp, 4702–4712 (2021)