Supplementary Material for UniTrack

Anonymous ICMR26 submission

Abstract

We provide the source code and additional visualization results at the following anonymous link: Anonymous GitHub Repository. The full implementation of UniTrack will be made publicly available upon acceptance. The code currently provided in the Github repository is also achieve the state-of-the-art performance compared with advanced TrackNet. The provided code repository is fully anonymized to strictly adhere to the double-blind review policy. We ensure that it contains no personally identifiable information (PII) regarding the authors. In this supplementary material, we provide comprehensive details regarding the implementation and architectural design of UniTrack. These additional materials are intended to facilitate a deeper understanding of the proposed framework and verify its effectiveness.

8 Motivation of Our UniTrack

8.1 Further Discussion on nmODEs

While high-performance U-shaped architectures—incorporating Transformers or complex attention mechanisms—yield superior tracking accuracy, their prohibitive parameter counts and computational costs (FLOPs) severely hinder real-world deployment [1, 7, 16]. Furthermore, existing lightweight solutions often necessitate holistic structural redesigns, resulting in highly coupled modules that lack transferability across different backbones. Consequently, developing a universal, lightweight decoding paradigm that preserves the encoder structure is of critical importance.

From a dynamical systems perspective, conventional Neural ODEs [2] typically initialize the system state directly with the external input (i.e., $y_0 = x$). However, strictly adhering to the ODE flow preserves homeomorphism, which constrains the network's capacity to map complex input spaces into semantic manifolds with distinct topological structures. To address this, we leverage the Neural Memory ODE (nmODE) [18]. Inspired by the dissociation between learning and memory in the cerebral cortex, nmODE models the decoding process as the evolution of a memory stream driven by external stimul, as shown in Fig. 1. The continuous dynamics are formulated as:

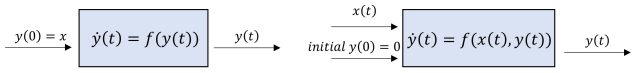
$$\dot{y}(t) = -y(t) + f\big(y(t) + g(x(t))\big) \tag{1}$$

where $x(t)$ denotes the external input at time t, and $y(t)$ represents the evolving memory flow. The term $-y(t)$ models state decay to ensure stability. $f(\cdot)$ serves as a non-linear mapping fusing existing memory with new inputs, while $g(\cdot)$ projects the external input $x(t)$ into an aligned feature space.

When adapted to the discrete architecture of a U-Net, this continuous process is instantiated as a layer-wise discrete stepping mechanism. For the i-th level, the update rule is governed by:

$$\dot{y}_i = -y_i + f\big(y_i + g(x_i)\big) \tag{2}$$

Here, x_i represents the input projection from the current encoder level (skip connection), and y_i denotes the corresponding decoder state. This formulation reformulates the cross-scale fusion in the decoder as the dynamical evolution of the nmODE system, achieving efficient bottom-up aggregation with minimal learnable parameters.



(a) ODE block (b) nmODE block

Figure 1: Comparison between traditional ODE and nmODE.

To specifically tailor the model for tiny-ball tracking, we refine the Initial Value Problem (IVP) of the dynamical system. Deviating from the standard zero-initialization ($y(0) = 0$), we utilize the output of the DDF module as the initial state. This strategy injects robust motion priors and a semantic foundation into the top-down propagation process, providing a solid physical basis for modeling the kinematics of tiny, fast-moving targets.

8.2 Further Discussion on LMM

Within the nmODE framework, the decoder state y_i is conceptualized as a continuous memory trajectory evolving on the feature manifold. The sequential encoder features $\{x_1, x_2, \ldots, x_n\}$ act as external forcing terms driving this system. Unlike traditional skip connections that rely solely on adjacent scales (instantaneous derivatives), we incorporate Linear Multistep Methods (LMMs) to explicitly aggregate information from multiple historical scales. This mechanism endows the network with high-order dynamical dependencies, enabling it to capture long-range spatiotemporal correlations beyond naive receptive field expansion.

While nmODE defines the local vector field $\dot{y} = \mathcal{F}(t, y)$, LMM serves as the high-order numerical integration operator. Formally, a general k-step linear multistep method is governed by the following theorem:

$$\sum_{j=0}^{N} \alpha_j \, y_{i+j} = \delta \sum_{j=0}^{N} \beta_j \, \mathcal{F}\big(t_{i+j}, \, y_{i+j}\big) \tag{3}$$

where δ denotes the discrete step size (scale interval), and α_j, β_j are method-specific coefficients. This formulation yields the explicit Adams–Bashforth scheme when $\beta_j = 0$, and the implicit Adams–Moulton scheme when $\beta_j \neq 0$. While implicit schemes are typically more accurate under the same step size, their updates depend on $\mathcal{F}(t_{i+j}, y_{i+j})$ at an unknown state, inducing an algebraic dependency that is non-trivial to resolve in feed-forward architectures. A standard remedy is the predictor–corrector paradigm [5, 6], which couples an explicit predictor with an implicit corrector.

Specifically, consider the IVP $\dot{y}(t) = F\big(t, y(t)\big)$ with $y(t_0) = y_0$ and step size δ. Given the current state $y_i \approx y(t_i)$, an explicit method first produces an initial prediction $\tilde{y}_{i+1}$, e.g.,

$$\tilde{y}_{i+1} = y_i + \delta \, F(x_i, y_i) \tag{4}$$

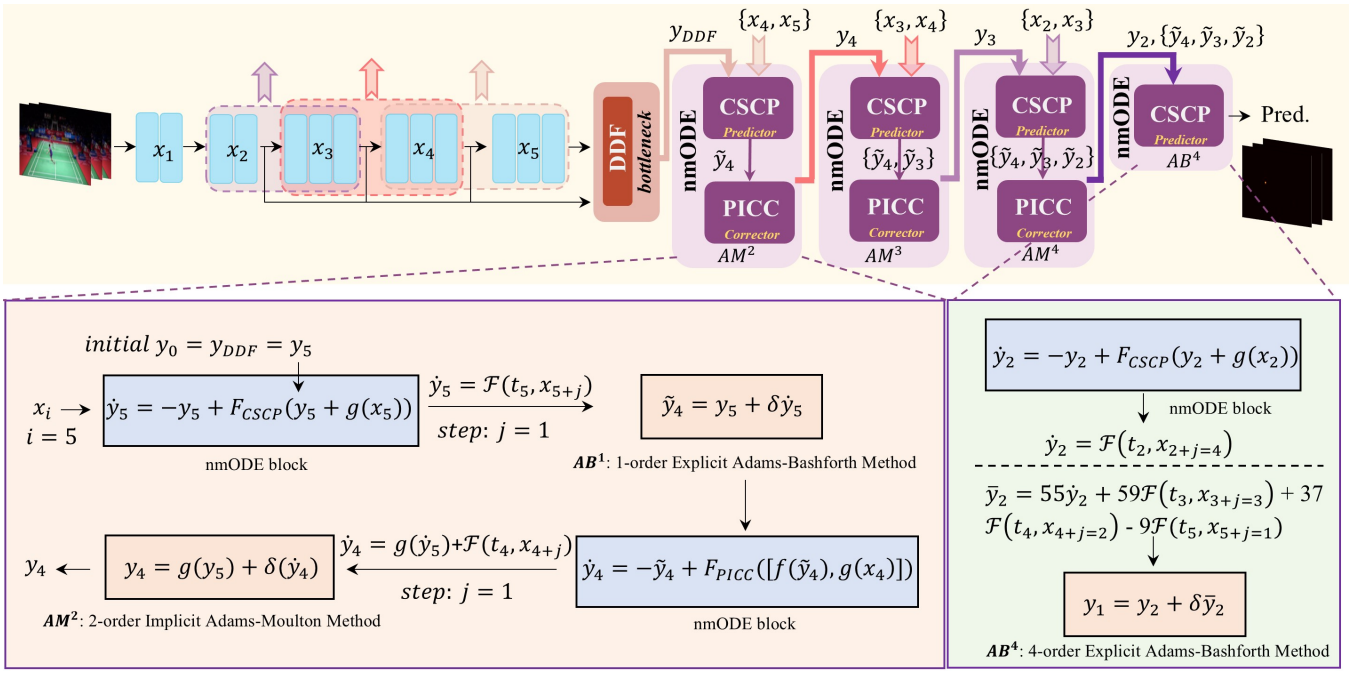


Figure 2: Detailed architecture of the proposed framework. We specifically illustrate the implicit calculation steps at stage $i = 5$ and the explicit calculation steps at stage $i = 2$ as representative examples.

Table 1: Linear multistep method. We denote $F_i := F(t_i, y_i)$ and δ as the step size.

Explicit Adams–Bashforth (AB)			Implicit Adams–Moulton (AM)		
Step	**Order**	**Update rule**	**Step**	**Order**	**Update rule**
1	1	$y_{i+1} = y_i + \delta F_i$	1	2	$y_{i+1} = y_i + \frac{\delta}{2}(F_i + F_{i+1})$
2	2	$y_{i+2} = y_{i+1} + \frac{\delta}{2}(3F_{i+1} - F_i)$	2	3	$y_{i+2} = y_{i+1} + \frac{\delta}{12}(5F_{i+2} + 8F_{i+1} - F_i)$
3	3	$y_{i+3} = y_{i+2} + \frac{\delta}{12}(23F_{i+2} - 16F_{i+1} + 5F_i)$	3	4	$y_{i+3} = y_{i+2} + \frac{\delta}{24}(9F_{i+3} + 19F_{i+2} - 5F_{i+1} + F_i)$
4	4	$y_{i+4} = y_{i+3} + \frac{\delta}{24}(55F_{i+3} - 59F_{i+2} + 37F_{i+1} - 9F_i)$	–	–	–

The Prediction then initializes the corresponding implicit update (e.g., the 2th-order Adams–Moulton AM^2 method):

$$y_{i+1} = y_i + \frac{\delta}{2}\Big(F(x_i, y_i) + F(x_{i+1}, \tilde{y}_{i+1})\Big) \tag{5}$$

To demonstrate how this process is specifically integrated into our decoder, Fig. 2 depicts the detailed workflow of our framework. Given an input frame, the encoder extracts a multi-scale feature pyramid $\{x_1, \ldots, x_5\}$. A DDF bottleneck produces the initial decoder state y_{DDF} (set as $y_0 = y_{\text{DDF}} = y_5$). We then perform top-down decoding by casting cross-scale fusion as an initial-value progression over scales. At each decoding stage, an nmODE block defines the continuous evolution.

$$\dot{y}_i = -y_i + F_{\text{CSCP}}\big(y_i + g(x_i)\big) \tag{6}$$

where $g(\cdot)$ aligning the skip feature to the state space. The next-scale state is obtained via a Predictor–Corrector scheme parameterized by LMM: an explicit Adams–Bashforth (AB) Predictor generates an intermediate estimate $\tilde{y}$, which initializes an implicit Adams–Moulton (AM) Corrector implemented by PICC, yielding the corrected state. For example, Starting from y_5, we first compute $\dot{y}_5$ via the nmODE. 1th-order AB Predictor (as shown in Table 1) advances one scale:

$$\tilde{y}_4 = y_5 + \delta\, \dot{y}_5 \qquad (\text{AB}^1) \tag{7}$$

The corrector then refines this estimate by coupling $\tilde{y}_4$ with the skip feature x_4:

$$\dot{y}_4 = -\tilde{y}_4 + F_{\text{PICC}}\big([\, f(\tilde{y}_4),\, g(x_4)\,]\big) \tag{8}$$

followed by an AM update (as shown in Table 1), producing the corrected y_4 (i.e., AM^2 at this stage). The same mechanism is applied stage-wise (i.e., AM^3 for y_3, AM^4 for y_2).

Table 2: Quantitative comparison on the Badminton benchmark, complementing the results in the main paper. The best two results are highlighted in red and blue.

Method	Param(M)	FPS	Badminton [13]			
			Acc.	Prec.	Rec.	F1
TrackNetV1 [8]	43.2	212.9	.8395	.9653	.8364	.8962
TrackNetV2 [13]	43.3	207.8	.8644	.9672	.8657	.9136
MonoTrack [9]	11.07	119.5	.8717	.9709	.8874	.9273
WASB [14]	5.7	77.9	.8446	.9469	.8585	.9006
MMFNet [19]	58.1	58.6	.9150	.9542	.9422	.9482
TrackNetV3 [3]	43.3	127.8	.9146	.9882	.9089	.9469
TOTNet [17]	8.7	26.3	.8899	.9549	.9092	.9315
BlurBall [4]	5.7	67.5	.8539	.9682	.8518	.9063
TrackNetV4 [12]	43.3	112.7	.9203	.9726	.9303	.9510
Yolo v7 [15]	71.3	148.75	.6131	.9455	.5605	.7038
DUTM [10]	50.6	18.1	.9090	.9707	.9178	.9435
STMENet [11]	9.85	24.25	.8833	.9727	.8844	.9624
Ours-light	35.8	182.6	.9352	.9748	.9469	.9606
Ours	58.0	85.7	.9414	.9750	.9540	.9644

Table 3: Additional quantitative analysis on other Badminton and Tennis benchmarks. The best two results are highlighted in red and blue.

Method	Badminton [20]				Tennis [20]			
	Acc.	Prec.	Recall	F1	Acc.	Prec.	Recall	F1
TrackNetV2	.8654	.9958	.8542	.9196	.8822	.9993	.8765	.9339
MonoTrack	.8773	.9936	.8695	.9274	.9231	.9976	.9212	.9579
WASB	.9089	.9861	.9117	.9474	.9395	.9863	.9495	.9675
TrackNetV3	.9342	.9790	.9473	.9629	.9551	.9872	.9652	.9761
MMFNet	.9418	.9849	.9499	.9671	.9508	.9884	.9594	.9737
TOTNet	.9285	.9920	.9282	.9590	.9443	.9981	.9431	.9698
BlurBall	.9254	.9923	.9245	.9572	.9401	.9891	.9474	.9678
TrackNetV4	.9492	.9814	.9619	.9715	.9570	.9850	.9694	.9772
Ours	.9668	.9850	.9781	.9815	.9748	.9881	.9853	.9867

To better exploit discrete historical nodes without incurring the complexity of higher-order implicit solving, we cap the implicit Adams–Moulton (AM) corrector at the 4th order. Once the decoding progression reaches a stage where the implicit order would exceed four, we uniformly switch the remaining transitions to a 4th-order explicit Adams–Bashforth (AB^4) update for simplicity and efficiency. As instantiated in Fig. 2 at stage i=2, since x_1 is not involved in decoding, it is omitted and the update leverages four available inputs $\{x_2, x_3, x_4, x_5\}$. We first evaluate the nmODE dynamics

$$\dot{y}_2 = -y_2 + F_{\text{CSCP}}\big(y_2 + g(x_2)\big) \tag{9}$$

and then apply the AB^4 Predictor (as shown in the figure) to aggregate multi-step derivative evaluations

$$\bar{y}_2 = 55\dot{y}_2 + 59F\big(t_3,\ x_{3+j=3}\big) + 37F\big(t_4,\ x_{4+j=2}\big) - 9F\big(t_5,\ x_{5+j=1}\big) \tag{10}$$

followed by the explicit update

$$y_1 = y_2 + \delta\,\bar{y}_2 \qquad (\text{AB}^4) \tag{11}$$

9 Extended Experimental Evaluation

9.1 Supplementary Quantitative Analysis

To complement the baselines shown in Fig. 1 but not reported in the quantitative table, we additionally benchmark three representative methods highlighted in gray in Table 2: (i) DUTM [10] and (ii) STMENet [11] (both originally proposed for moving infrared small-target tracking, representing implicit motion-direction encoding/multi-frame directional consistency modeling and decoupled 2D–3D spatio-temporal representation learning, respectively), and (iii) YOLOv7-X [15], an efficient single-frame detector used to estimate the performance ceiling of the detection-only paradigm in fast ball tracking. Specifically, DUTM introduces a Direction-Coded Convolution Block (DCCB) within a temporal U-shaped architecture to implicitly encode motion directions and exploit multi-frame directional consistency for target enhancement and clutter suppression. STMENet adopts a decoupled 2D–3D dual-branch encoder, where the 3D backbone captures inter-frame motion cues, the 2D backbone models multi-scale spatial appearance, and a Spatial-Temporal Mix Encoder (STME) enables learnable channel-wise interactions for spatio-temporal fusion. For YOLOv7-X, we investigate whether an efficient detection model can effectively handle fast-moving ball tracking. We therefore train it end-to-end from scratch on our badminton dataset and adapt the output head for ball localization. Meanwhile, all newly added baselines strictly follow the same dataset split, training protocol, and evaluation settings as in the main paper to ensure fair comparison.

Table 2 demonstrates that these three design choices—direction priors, local 3D convolutions, and single-frame detection—remain notably limited for high-speed badminton ball tracking. Although single-frame detection achieves high throughput (YOLOv7: 148.75 FPS), it suffers severe missed detections under strong motion blur and extremely small scales, yielding a Recall of only 0.5605 and reducing F1 to 0.7038. Direction encoding provides some gains (DUTM: F1=0.9435), but it still lags behind our method (Ours: F1=0.9644) and runs at only 18.1 FPS, making it difficult to balance accuracy and efficiency. This gap is closely tied to its underlying assumptions: infrared small-target data often exhibit near-linear, unidirectional motion, whereas a badminton shuttlecock undergoes abrupt, highly nonlinear direction changes after impact, making direction encoding prone to mismatch and error accumulation during temporal aggregation. 3D spatiotemporal modeling (STMENet) further improves spatiotemporal representation (F1=0.9624), but it remains prone to missed detections or drift under short-term disappearance caused by extreme blur, confusion with background-like distractors. Overall, relying solely on directional priors or stacking 3D convolutions is insufficient to robustly handle abrupt direction changes and imaging degradation in high-speed small-ball

scenarios; an end-to-end tracking architecture with multi-scale representations and spatiotemporal fusion remains a more effective path. Inspired by the use of multi-scale features for small-target localization in MonoTrack, MMFNet, and DUTM, we integrate this idea into the Key branch of the CSCP module, achieving more robust tiny-ball tracking in complex scenarios and achieving SOTA performance.

9.2 Additional Quantitative Analysis

Table 3 further validates the cross-benchmark generalization of our approach on two additional badminton and tennis benchmarks [20], where we achieve the best overall performance on both. Specifically, we obtain 96.68% Acc., 98.15% F1 on Badminton, and 97.48% Acc., 98.67% F1 on Tennis. Compared with the strongest baseline TrackNetV4, our method improves F1 by 1.03%/0.97% and Acc. by 1.85%/1.86%, together with consistent recall gains of 1.68%/1.64%. Importantly, these gains are achieved while maintaining high precision (Prec.=0.37%/0.31%), avoiding the recall drop and F1 degradation caused by overly conservative predictions (e.g., TrackNetV2).

In addition, we observe a consistent performance degradation for all methods on the tennis benchmark used in the main paper compared with the numbers reported in the original publications. We attribute this discrepancy primarily to differences in data accessibility and quality: the official webpage of the original tennis dataset is no longer available, and the currently accessible version is sourced from WASB and distributed as extracted video frames, which may have undergone repeated transcoding/compression and frame extraction, leading to reduced image fidelity and potential frame drops. To mitigate this factor, we further evaluate on another publicly available tennis tracking benchmark and retrain all methods end-to-end from scratch under exactly the same configuration as in the main paper. Looking forward, we plan to build and release a unified benchmark covering multiple ball types to facilitate more reproducible and diverse evaluations.

9.3 Additional Visualization Results

To further emphasize the localization accuracy of our LMM-based decoding architecture, we provide extensive visualizations of stage-wise heatmaps and their overlays on the original frames, focusing exclusively on our method without side-by-side comparisons, as shown in Fig. 3. Across diverse and highly challenging cases (e.g., severe motion blur, cluttered backgrounds, and partial/short-term occlusions), the response maps exhibit a stable coarse-to-fine evolution over the four decoding stages: the peak remains consistently aligned with the ball center and becomes progressively sharper as decoding proceeds, while spurious activations around nearby distractors are gradually suppressed. This behavior indicates that the iterative Predictor–Corrector refinement can continuously propagate reliable cues and rectify ambiguous observations, thereby producing accurate and robust ball localization in complex scenes.

Building upon the qualitative results in the main paper, we further extend the visualization study by presenting more tracking examples on Badminton, Table Tennis, and Tennis benchmarks, as shown in Fig. 4-5. We include additional badminton sequences featuring abrupt trajectory changes after hits, extremely small targets, and heavy background interference, together with representative table-tennis and tennis clips that involve fast motion, specular highlights, scale variations, and long-range displacement. Across these scenarios, our method maintains temporally coherent trajectories and precise frame-wise localization under motion blur, occlusions, and visually similar distractors, demonstrating strong cross-dataset generalization of the proposed LMM-based decoder.

References

[1] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. 2022. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*. Springer, 205–218.

[2] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems* 31 (2018).

[3] Yu-Jou Chen and Yu-Shuen Wang. 2023. Tracknetv3: Enhancing shuttlecock tracking with augmentations and trajectory rectification. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia*. 1–7.

[4] Thomas Gossard, Filip Radovic, Andreas Ziegler, and Andrea Zell. 2025. Blurball: Joint ball and motion blur estimation for table tennis ball tracking. *arXiv preprint arXiv:2509.18387* (2025).

[5] William B Gragg and Hans J Stetter. 1964. Generalized multistep predictor-corrector methods. *Journal of the ACM (JACM)* 11, 2 (1964), 188–209.

[6] Karl Heun et al. 1900. Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. *Z. Math. Phys* 45 (1900), 23–38.

[7] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. 2020. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Ieee, 1055–1059.

[8] Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsì-Uí Îk, and Wen-Chih Peng. 2019. 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 (AVSS)*. IEEE, 1–8.

[9] Paul Liu and Jui-Hsien Wang. 2022. MonoTrack: Shuttle trajectory reconstruction from monocular badminton video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3513–3522.

[10] Dengyan Luo, Yanping Xiang, Hu Wang, Luping Ji Shuai Li, and Mao Ye. 2025. Bidirectional Temporal Information Propagation for Moving Infrared Small Target Detection. *arXiv preprint arXiv:2508.15415* (2025).

[11] Shuang Peng, Luping Ji, Shengjia Chen, Weiwei Duan, and Sicheng Zhu. 2025. Moving infrared dim and small target detection by mixed spatio-temporal encoding. *Engineering Applications of Artificial Intelligence* 144 (2025), 110100.

[12] Arjun Raj, Lei Wang, and Tom Gedeon. 2025. Tracknetv4: Enhancing fast sports object tracking with motion attention maps. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[13] Nien-En Sun, Yu-Ching Lin, Shao-Ping Chuang, Tzu-Han Hsu, Dung-Ru Yu, Ho-Yi Chung, and Tsì-Uí Îk. 2020. Tracknetv2: Efficient shuttlecock tracking network. In *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*. IEEE, 86–91.

[14] Shuhei Tarashima, Muhammad Abdul Haq, Yushan Wang, and Norio Tagawa. 2023. Widely applicable strong baseline for sports ball detection and tracking. *arXiv preprint arXiv:2311.05237* (2023).

[15] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7464–7475.

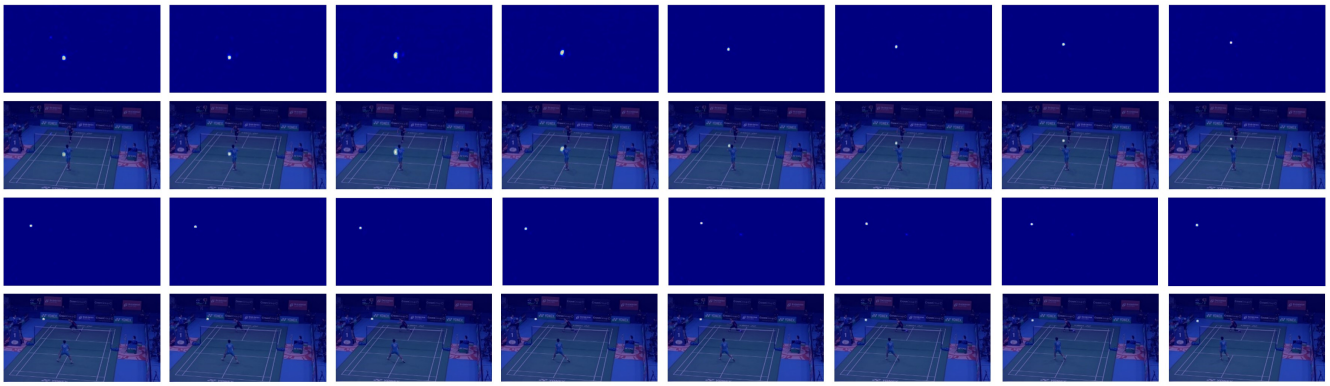
[16] Xinyu Xiong, Zihuang Wu, Shuangyi Tan, Wenxue Li, Feilong Tang, Ying Chen, Siying Li, Jie Ma, and Guanbin Li. 2026. Sam2-unet: Segment anything 2 makes strong encoder for natural and medical image segmentation. *Visual Intelligence* 4, 1 (2026), 2.

[17] Hao Xu, Arbind Agrahari Baniya, Sam Wells, Mohamed Reda Bouadjenek, Richard Dazeley, and Sunil Aryal. 2026. TOTNet: Occlusion-aware temporal tracking for robust ball detection in sports videos. *Computer Vision and Image Understanding* (2026), 104657.

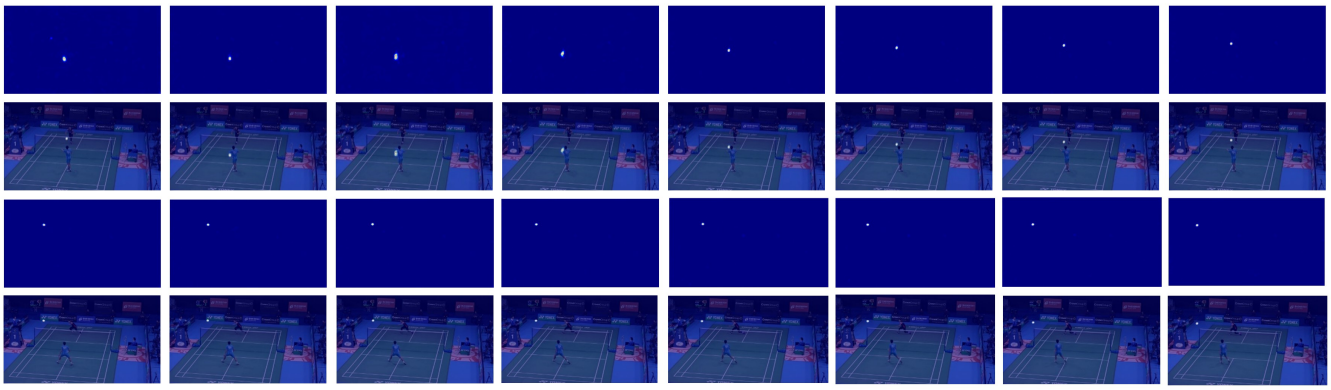
[18] Zhang Yi. 2023. nmODE: neural memory ordinary differential equation. *Artificial Intelligence Review* 56, 12 (2023), 14403–14438.

[19] Jizhe Yu, Yu Liu, Hongkui Wei, and Kaiping Xu. 2024. Towards More Accurate Tiny Object Tracking: Benchmark and Algorithm. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 1858–1863.

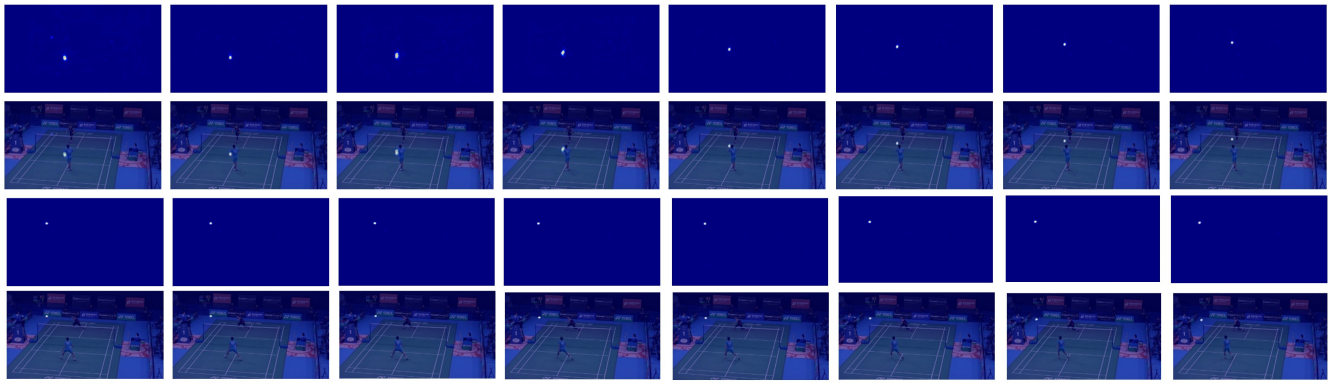
[20] Jizhe Yu, Yu Liu, Hongkui Wei, Kaiping Xu, Yifei Cao, and Jiangquan Li. 2024. Towards Highly Effective Moving Tiny Ball Tracking via Vision Transformer. In



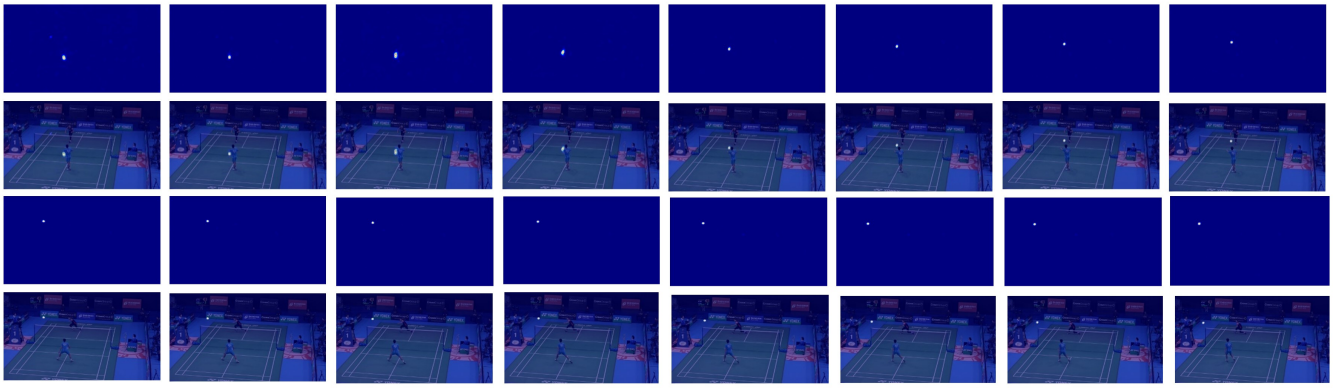
(a) 4-stage of Decoder



(b) 3-stage of Decoder

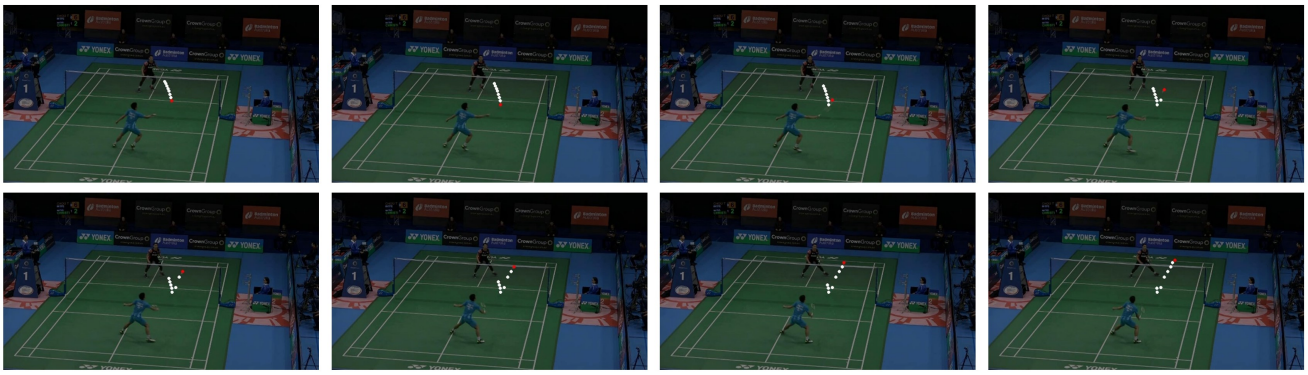


(c) 2-stage of Decoder

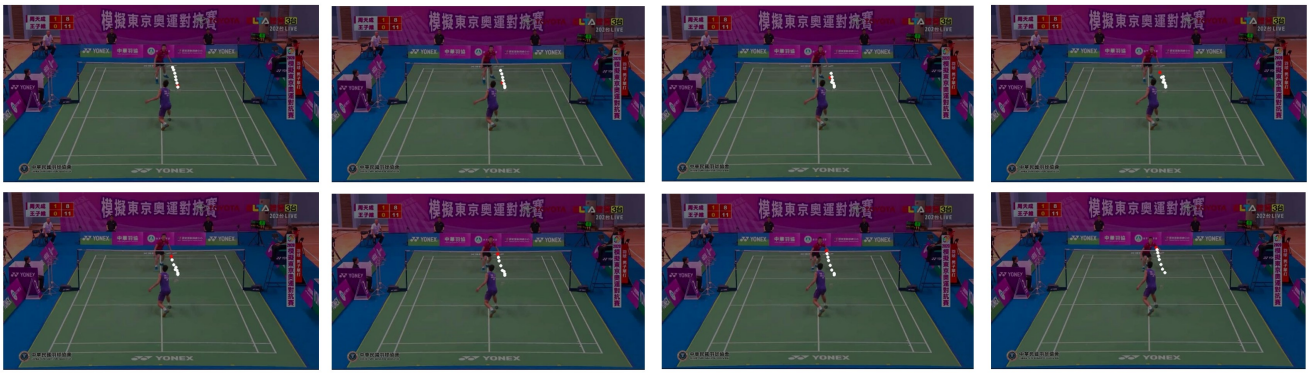


(d) 1-stage of Decoder

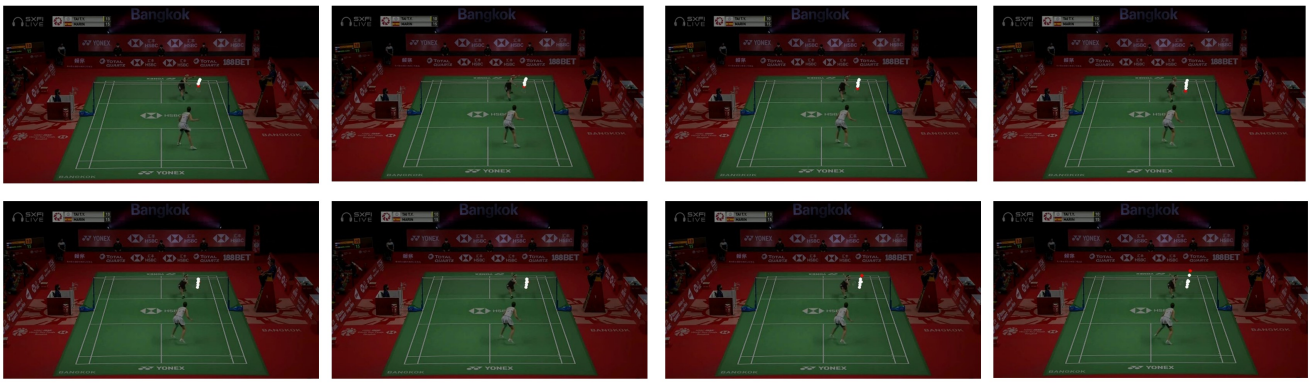
Figure 3: Stage-wise visualizations of our LMM-based decoder. We show the predicted heatmaps at four decoding stages and their overlays on the original frames. Across challenging cases (e.g., motion blur, clutter, and occlusion), the response peak remains aligned with the ball center and becomes progressively sharper, illustrating a stable coarse-to-fine Predictor–Corrector refinement.



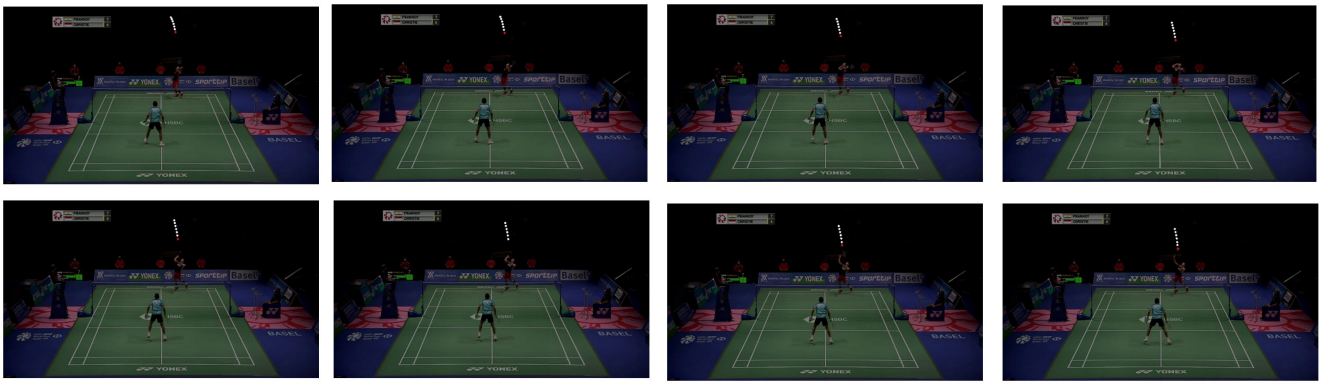
(a) Case one



(b) Case two



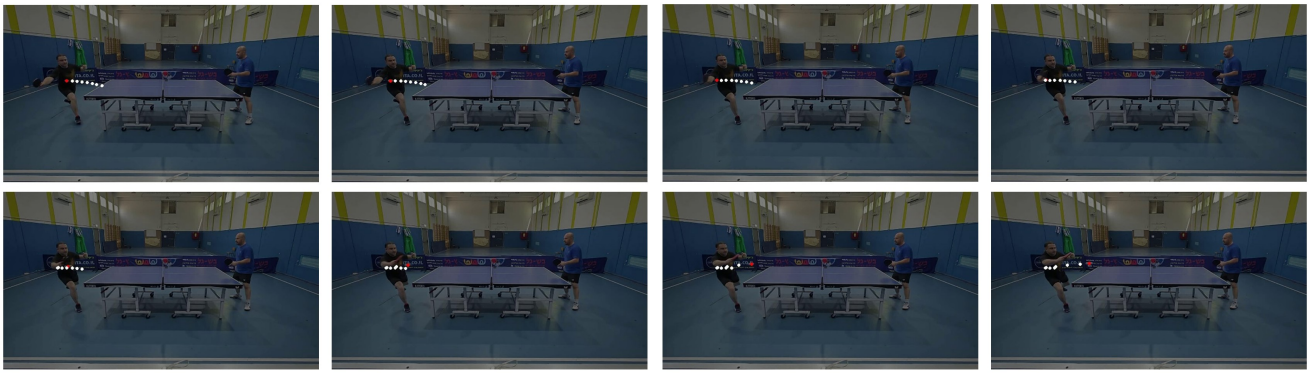
(c) Case three



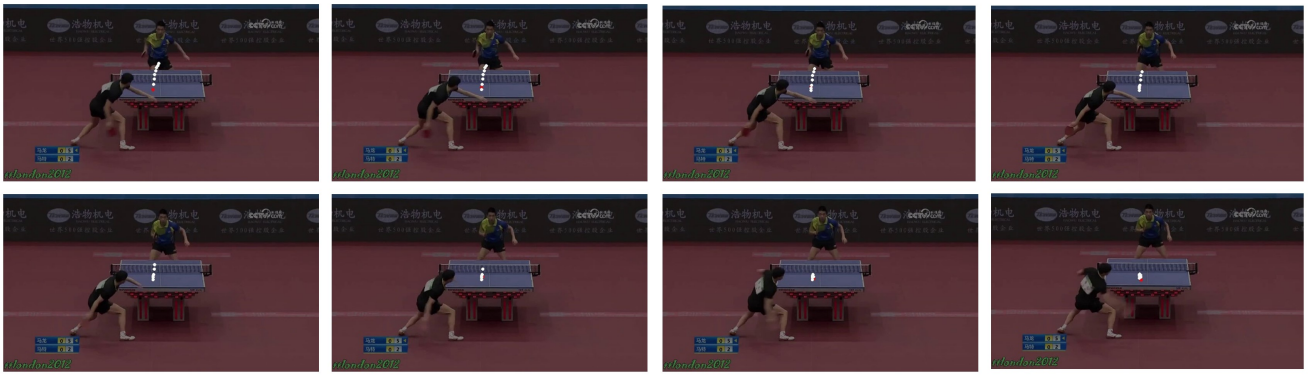
(c) Case four

Figure 4: Visualization of ball trajectories on the test set of Badminton tracking dataset.

(a) Case one



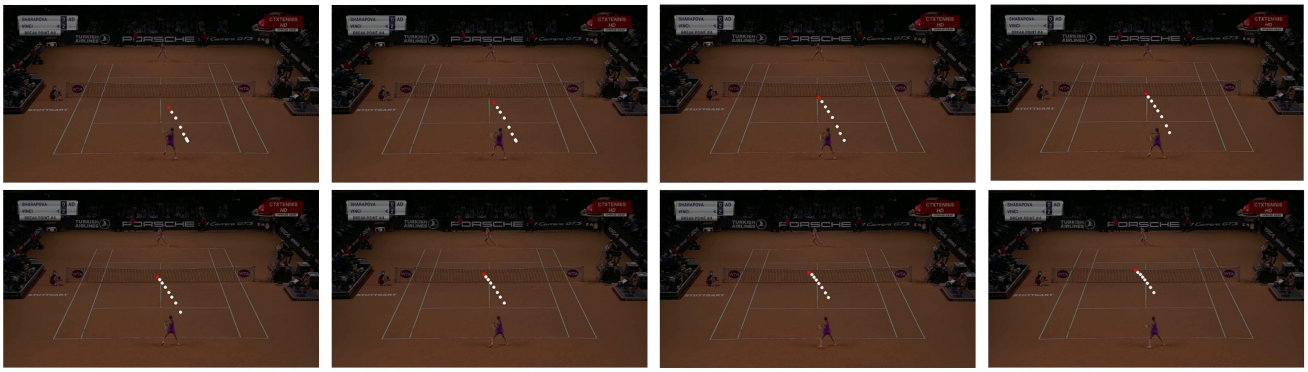
(b) Case two



(c) Case three

(d) Case four

Figure 5: Visualization of ball trajectories on the test set of Table tennis ball tracking dataset.



(a) Case one

(b) Case two

(c) Case three

(d) Case four

Figure 6: Visualization of ball trajectories on the test set of Tennis ball tracking dataset.

International Conference on Intelligent Computing. Springer, 368–379.