# Multi-view Crowd Tracking Transformer with View-Ground Interactions Under Large Real-World Scenes Supplemental

Qi Zhang[1] Jixuan Chen[1] Kaiyi Zhang[1] Xinquan Yu[1]
Antoni B. Chan[2] Hui Huang[1*]
[1]College of Computer Science and Software Engineering, Shenzhen University, China
[2]Department of Computer Science, City University of Hong Kong, China
{qi.zhang.opt, chen.jixuanstu, zhangky1999, xinquanyu2619}@gmail.com,
abchan@cityu.edu.hk, hhzhiyan@gmail.com

## 1. More Details

### 1.1. Dataset Details

**Dataset Annotation.** Our MVCrowdTrack and CityTrack datasets provide frame-level annotations, including 2D bounding boxes and ground-plane coordinates in a unified BEV coordinate system. To enable efficient and accurate labeling, we developed a customized *multi-camera annotation tool* that allows annotators to visualize and adjust object positions across all camera views simultaneously. Professional annotators were invited to complete the labeling using this tool. After annotation, all samples were carefully filtered and manually inspected to ensure high accuracy and consistency across views.

**Camera Calibration.** For each camera, we first calibrate the intrinsic parameters using the standard MATLAB chessboard calibration toolkit. To obtain accurate extrinsic parameters, we perform a coarse initialization using a PnP-based procedure: we manually select corresponding points between each camera image and satellite maps to estimate an initial camera pose. Afterward, we apply a multi-camera joint refinement by leveraging annotated person locations. Specifically, we use the head and foot positions of annotated pedestrians and assume a human height of 170 cm to optimize the camera poses across all views in a second-stage bundle adjustment. This two-step calibration process leads to stable and geometrically consistent extrinsic parameters across the entire multi-camera system.

**MVCrowdTrack Dataset.** To support large-scale and realistic multi-view crowd tracking, we construct the *MVCrowdTrack* dataset captured in a 120 m $\times$ 80 m outdoor campus environment. Seven high-resolution cameras (5312$\times$2988) simultaneously record the scene for 18 minutes at 60 fps. We annotate the videos at 4 fps, producing a total of 4,122 multi-view frames. The dataset contains 342 person trajectories with an average duration of 176 frames. We use 3,297 frames for training and 825 for testing. Annotators provide per-view bounding boxes with consistent identities, and calibrated camera parameters project visible foot points onto the ground plane, where their average is taken as the BEV annotation. The final BEV map has a resolution of 1200$\times$800 with a spatial scale of 0.1 m per pixel. Compared with prior datasets, MVCrowdTrack offers significantly larger spatial coverage, higher image resolution, denser temporal annotations, and longer trajectories.

**CityTrack Dataset.** CityTrack is constructed by re-annotating part of the CityStreet dataset for multi-view crowd tracking. It contains 2,588 frames sampled at 4 fps, with identity-consistent annotations across time. The dataset provides long-term trajectories with an average length of 228 frames. We adopt the first 1,948 frames for training and the remaining 640 for testing. The ground-plane resolution is 640$\times$768, corresponding to 0.1 m per pixel. Compared with previous benchmarks, CityTrack offers denser temporal sampling and higher crowd density, making it suitable for evaluating multi-view tracking models in busy urban scenes.

### 1.2. Evaluation Metrics

All tracking results are evaluated on the ground plane to ensure a unified spatial coordinate system across datasets. Following the standard protocol in recent multi-view tracking works, we report both accuracy-based [**?** ] and identity-aware [3] metrics. A predicted trajectory point is considered correctly matched to a ground-truth location if their BEV distance is within a dataset-specific association radius: we use a threshold of 2 m for the large-scale *MVCrowdTrack* and *CityTrack* datasets, and 1 m for *Wildtrack* and *MultiviewX* where scenes are more compact.

* Corresponding author

**MOTA.** Multiple Object Tracking Accuracy (MOTA) measures overall tracking quality by penalizing false positives (FP), false negatives (FN), and identity switches (IDSW):

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FN} + \mathrm{FP} + \mathrm{IDSW}}{\mathrm{GT}}, \tag{1}$$

where GT denotes the total number of ground-truth annotations.

**MOTP.** Multiple Object Tracking Precision (MOTP) evaluates localization precision by computing the average BEV distance between matched prediction–GT pairs:

$$\mathrm{MOTP} = \frac{\sum_t \sum_{i \in \mathcal{M}_t} d_{t,i}}{\sum_t |\mathcal{M}_t|}, \tag{2}$$

where $d_{t,i}$ is the BEV distance of the $i$-th match at time $t$, and $\mathcal{M}_t$ denotes the set of matched pairs in frame $t$.

**IDF1, IDP, and IDR.** Identity-based metrics evaluate whether a tracker preserves person identities over time. Identity precision and recall are defined as:

$$\begin{aligned} \mathrm{IDP} &= \frac{\mathrm{IDTP}}{\mathrm{IDTP} + \mathrm{IDFP}}, \\ \mathrm{IDR} &= \frac{\mathrm{IDTP}}{\mathrm{IDTP} + \mathrm{IDFN}}. \end{aligned} \tag{3}$$

where IDTP, IDFP, and IDFN denote identity true positives, false positives, and false negatives, respectively. Identity F1 score is then computed as:

$$\mathrm{IDF1} = \frac{2 * \mathrm{IDP} * \mathrm{IDR}}{\mathrm{IDP} + \mathrm{IDR}}. \tag{4}$$

**MT and ML.** We further evaluate trajectory completeness using Mostly Tracked (MT) and Mostly Lost (ML). An estimated trajectory is considered MT if it is tracked for at least 80% of its lifespan, and ML if tracked for less than 20%. Formally, letting "#" denote the number of elements:

$$\begin{aligned} \mathrm{MT} &= \frac{\#\{\, \frac{T^{\mathrm{tracked}}}{T^{\mathrm{total}}} \geq 0.8 \,\}}{\#\text{GT trajectories}}, \\ \mathrm{ML} &= \frac{\#\{\, \frac{T^{\mathrm{tracked}}}{T^{\mathrm{total}}} \leq 0.2 \,\}}{\#\text{GT trajectories}}. \end{aligned} \tag{5}$$

These metrics measure long-term association stability and are particularly informative in crowded multi-view environments.

### 1.3. More Implementation and Training Details

During training, we apply random resizing and cropping to the multi-view image sequences as the main data augmentation strategy. The input images are randomly scaled within the range of 0.8–1.2 and randomly cropped. For all datasets, the resized input resolution is fixed to $720 \times 1280$.

We train the model for 50 epochs using the AdamW optimizer with a learning rate of 0.01, along with a cosine learning rate scheduler with warm-up. The hidden dimension of the Transformer layers is set to 64, and the feed-forward network (FFN) dimension is set to 512. A dropout rate of 0.1 is applied to the attention layers.

For the tracking task, the model takes two consecutive frames as input. The forward pass of the previous frame is frozen, while gradients are enabled only for the current frame to stabilize temporal training.

To ensure a fair comparison, all methods in our experiments adopt the same BEV grid resolution. The grid sizes for each dataset are:

- **MultiviewX**: $160 \times 250$
- **Wildtrack**: $120 \times 360$
- **CityTrack**: $384 \times 320$
- **MVCrowdTrack**: $200 \times 300$

As both TrackTacular and our model utilize 3D features, we also align the BEV height dimension and set it to 4 across all datasets.

**Computational Resource.** All experiments are conducted on a workstation equipped with four NVIDIA RTX 4090 GPUs. We train our model in a data-parallel manner across all four GPUs. On the largest setting—the *MVCrowdTrack* dataset with seven camera views and a BEV grid resolution of $200 \times 300$—the peak memory usage per GPU is approximately 12.46 GB. This configuration comfortably fits within the available resources and allows stable end-to-end training of both the ground-view and multi-view branches.

## 2. More Experiments

### 2.1. Ablation study on improving temporal consistency with kalman filtering

In our main framework, the tracking module relies solely on the Transformer decoder to predict motion offsets between consecutive frames. However, this design results in noticeably lower IDF1 scores on MultiviewX, which emphasizes long-term identity consistency over complete trajectories.

To mitigate this issue, we integrate a Kalman Filter into the tracking pipeline to enhance temporal state estimation and smoothing. The Kalman Filter maintains a per-target

Table 1. Ablation study on improving temporal consistency with kalman filtering on MultiviewX dataset. Integrating Kalman filtering into our method leads to notable gains in both MOTA and IDF1.

| Method | MOTA↑ | MOTP↑ | IDF1↑ | MT↑ | ML↓ |
|---|---|---|---|---|---|
| REMP$^{\dagger}$ [1] | 81.0 | 85.8 | – | – | – |
| EarlyBird [5] | 88.4 | 86.2 | 82.4 | 82.9 | 1.3 |
| TrackTacular [4] | 92.4 | 80.1 | 85.6 | 92.1 | 2.6 |
| MVTr [7] | 91.4 | **95.0** | 82.9 | 96.1 | **0.0** |
| MVTrajecter [6] | **92.8** | **95.0** | **85.8** | **97.4** | **0.0** |
| MVTrackTrans (Ours) | 89.8 | 90.2 | 72.1 | 85.5 | 6.6 |
| MVTrackTrans + Kalman filter (Ours) | 91.0 | 83.7 | 84.1 | 92.1 | 2.7 |

Table 2. Ablation study on different numbers of views on the Wildtrack dataset shows that performance consistently decreases as fewer camera views are available, while the best results are achieved when all views are utilized.

| Variant | MOTA↑ | MOTP↑ | IDF1↑ | MT↑ | ML↓ |
|---|---|---|---|---|---|
| 3 cams | 81.4 | 84.3 | 87.3 | 56.1 | 9.8 |
| 4 cams | 83.4 | 83.2 | 88.7 | 65.9 | 7.3 |
| 5 cams | 85.8 | 84.3 | 89.3 | 70.7 | 7.3 |
| 6 cams | 90.5 | 83.2 | 90.3 | 73.2 | **4.9** |
| 7 cams | 87.6 | 85.3 | 91.7 | 70.7 | 9.8 |
| Full Model (7 cams) | **91.2** | **86.9** | **94.1** | **82.9** | **4.9** |

motion model and updates each state by fusing the predicted offsets with historical observations.

As shown in Table 1, incorporating Kalman filtering produces more stable ground-plane trajectories and reduces identity fragmentation, leading to clear improvements in both MOTA and IDF1 on the MultiviewX dataset.

## 2.2. Ablation study on different numbers of views

To evaluate the robustness of our multi-view feature aggregation, we replace the original $1 \times 1$ convolution-based fusion with a max-pooling operation, which naturally supports a variable number of input cameras. This modification removes the constraint of a fixed camera count at test time, enabling flexible evaluation under different view configurations. The model is trained using all available camera views on the Wildttrack dataset, and during testing, we progressively drop cameras to assess how the number of active views influences tracking performance.

From Table 2, we observe the following: The last row reports the performance of the original $1 \times 1$ convolution-based fusion trained and tested with all views, which achieves the best scores across all metrics. In the remaining rows, the convolution is replaced with max-pooling, and cameras are progressively removed at test time. As expected, performance decreases as fewer views are available, reflecting the direct impact of view count on tracking accuracy. Nevertheless, the max-pooling aggregation demonstrates strong robustness under varying numbers of input cameras, exhibiting only moderate declines in MOTA and IDF1. This confirms its suitability for scenarios with dynamic or incomplete camera configurations.

Table 3. Tracking performance comparison on the GMVD dataset.

| Method | IDF1↑ | MOTA↑ | MOTP↑ | MT↑ | ML↓ |
|---|---|---|---|---|---|
| EarlyBird | 69.70 | 70.80 | 86.10 | 41.60 | 12.40 |
| MVFlow | 70.50 | 71.10 | 85.70 | 41.00 | 12.70 |
| TrackTacular | 74.40 | 71.20 | 85.20 | 52.40 | 12.50 |
| MVTr | 74.30 | 75.60 | 84.20 | 63.10 | 10.30 |
| MvTrajecter | 77.20 | 78.10 | **87.70** | **66.00** | 8.70 |
| MVTrackTrans (Ours) | **78.60** | **78.16** | 84.97 | 56.23 | **8.50** |

Table 4. Additional metrics comparison on the CityTrack and MVCrowdTrack dataset.

| Dataset | CityTrack | | | MVCrowdTrack | | |
|---|---|---|---|---|---|---|
| Method | HOTA↑ | DetA↑ | AssA↑ | HOTA↑ | DetA↑ | AssA↑ |
| EarlyBird | 33.48 | 56.64 | 17.21 | 52.17 | 60.15 | 44.29 |
| TrackTacular | 34.75 | 47.45 | 22.12 | 61.21 | **73.57** | 48.77 |
| MVFlow | 35.08 | 40.08 | **27.28** | 61.35 | 68.45 | **53.49** |
| **Ours** | **37.62** | **61.51** | 19.97 | **61.63** | 71.71 | 50.92 |

## 2.3. Additional Dataset Evaluations.

To further evaluate the generalization ability of our method, we conduct additional experiments on the GMVD dataset. The comparison results are shown in Table 3. Our method achieves the best performance on most metrics, including IDF1 and MOTA, outperforming previous approaches. These results demonstrate that MVTrackTrans maintains strong tracking performance and robustness on additional datasets.

## 2.4. Additional Experimental Metrics

Although HOTA and related metric [**?** ] were not included in the main comparison since some previous methods did not report them in their original papers, they are important evaluation metrics in tracking. Therefore, we additionally report HOTA, DetA, and AssA in Table 4. Our method achieves the best HOTA on both datasets, further validating the effectiveness of our approach.

## 2.5. Computation cost.

As our MVTrackTrans model adopts a Transformer-based architecture and requires two consecutive frames as input during training, the overall number of parameters and intermediate feature maps is naturally larger than single-frame or non-transformer counterparts. This leads to a slightly higher GPU memory footprint. Nevertheless, as shown in Table 5, adding the View Prediction branch increases the computation cost only slightly. Building on this, introducing our *View–Ground Interaction Module* adds only a marginal amount of additional memory usage (from 9.962,GB to 9.968,GB). This tiny difference (less than 0.1%) indicates that the proposed interaction module is lightweight and imposes almost no additional overhead during training.

Furthermore, Table 6 compares our method with other state-of-the-art multi-view approaches. Although our Transformer-based design and multi-frame training lead to slightly larger FLOPs, the testing-time latency remains competitive. One possible reason for the differences in

Table 5. Computation cost comparison among different variants of our proposed MVTrackTrans model. (All memory usage and FLOPs are measured during training.)

| Loss | Memory(GB) | FLOPs(G) | Train(s) | Test(s) |
|---|---|---|---|---|
| Baseline | 9.036 | 1155.654 | 0.873 | 0.098 |
| + View Prediction Branch | 9.962 | 1288.654 | 0.901 | 0.110 |
| ++ ViewInteraction (Ours) | 9.968 | 1350.771 | 0.943 | 0.118 |

Table 6. Computation cost comparison among different methods. (All memory usage and FLOPs are measured during training.)

| Method | Memory(GB) | FLOPs(G) | Train(s) | Test(s) |
|---|---|---|---|---|
| EarlyBird [5] | 5.880 | 1074.787 | 0.314 | 0.064 |
| TrackTacular [4] | 9.838 | 1221.315 | 0.298 | 0.068 |
| MVFlow [2] | 8.717 | 1055.347 | 0.424 | 0.075 |
| MVTrackTrans (Ours) | 9.968 | 1350.771 | 0.943 | 0.118 |

reported training speed is that some prior works such as EarlyBird [5] and TrackTacular [4] measure runtime using `PyTorch Lightning` [**?** ], which typically results in shorter recorded training times.

## 3. More Visualization Results

We provide additional visualization results on the MVCrowdTrack dataset in Figure 1. As highlighted by the red boxes, our method delivers more reliable tracking, exhibiting fewer missed detections and producing smoother, more consistent trajectories. In contrast, competing approaches either generate false positives (TrackTacular [4]) or suffer from identity switches (MVFlow [2]).

## 4. Ethical Discussion

Our MVCrowdTrack dataset was collected in outdoor public areas on a university campus with official permission. The campus is equipped with cameras for security reasons, where the individuals are informed of the existence of the cameras, and no privacy invasion is conducted. Specifically, no personal identifiers, demographic labels, audio, or sensitive metadata were collected. To prevent potential misuse for face recognition, all visible faces are masked before release, and only tracking-related positional information is retained.

Although the dataset reflects benign crowd-analysis scenarios, multi-camera tracking technology could, in principle, be misused for intrusive surveillance. To reduce such risks, the dataset is released strictly for non-commercial academic research, contains no attributes that enable profiling or discrimination, and complies with common data-protection expectations. Individuals cannot be meaningfully re-identified after anonymization, and removal requests can be processed through the university authority that approved the recording.

Overall, our goal is to support research in multi-view crowd understanding while remaining transparent about potential risks. Through face anonymization, restricted release conditions, and the exclusion of sensitive features, we aim to minimize harmful use. We encourage downstream researchers to carefully consider privacy, fairness, and application-level implications when deploying related systems.

## References


[1] Kosta Dakic, Kanchana Thilakarathna, Rodrigo N. Calheiros, and Teng Joon Lim. Resource-efficient multiview perception: Integrating semantic masking with masked autoencoders. In *2025 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 145–151, 2025. 3

[2] Martin Engilberge, Weizhe Liu, and Pascal Fua. Multi-view tracking using weakly supervised human motion prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1582–1592, 2023. 4

[3] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35, 2016. 1

[4] Torben Teepe, Philipp Wolters, Johannes Gilg, Fabian Herzog, and Gerhard Rigoll. Lifting multi-view detection and tracking to the bird's eye view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 667–676, 2024. 3, 4

[5] Torben Teepe, Philipp Wolters, Johannes Gilg, Fabian Herzog, and Gerhard Rigoll. Earlybird: Early-fusion for multi-view tracking in the bird's eye view. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 102–111, 2024. 3, 4

[6] Taiga Yamane, Ryo Masumura, Satoshi Suzuki, and Shota Orihashi. Mvtrajecter: Multi-view pedestrian tracking with trajectory motion cost and trajectory appearance cost. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13270–13280, 2025. 3

[7] Yihan Yang, Ming Xu, Jason F Ralph, Yuchen Ling, and Xiaonan Pan. An end-to-end tracking framework via multi-view and temporal feature aggregation. *Computer Vision and Image Understanding*, 249:104203, 2024. 3

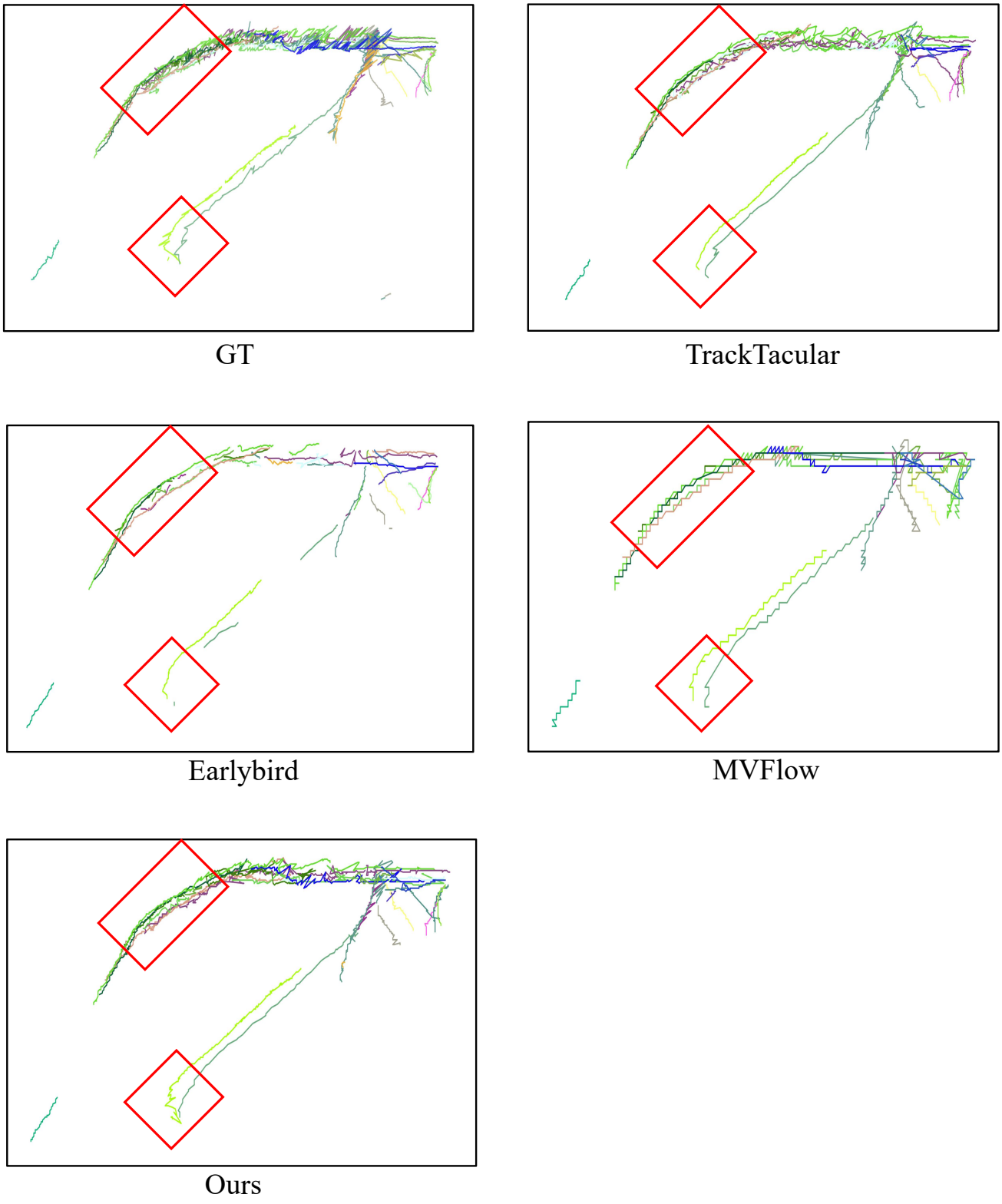


Figure 1. The predicted trajectory visualizations on the MVCrowdTrack dataset. Our method can accurately track more people for a long time (see red boxes). The visualization corresponds to frames 3200–3800 of the dataset.