MITracker: Multi-View Integration for Visual Object Tracking

Supplementary Material

Section 9 provides additional information on the MVTrack dataset, while Section 10 includes further implementation details and experimental results of MITracker.

9. Dataset Details

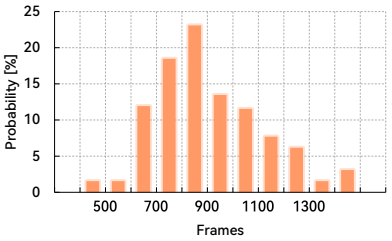
Data Annotation. In the BEV annotations, the MVTrack dataset covers an $8m \times 8m$ area. Ground truth labels are projected onto a 400×400 grid, where each cell is $2cm \times 2cm$ in size.

Attributes Definition. MVTrack dataset contains nine attributes to assess tracking robustness, as shown in Table 4. We provide frame-level binary labels for five attributes: Background Clutter (BC), Motion Blur (MB), Partial Occlusion (POC), Full Occlusion (FOC), and Out of View (OV). These are manually annotated for each frame. Deformation (DEF) is labeled according to whether the tracked target deforms. Low Resolution (LR), Aspect Ratio Change (ARC), and Scale Variation (SV) are automatically computed from changes in the BBox size.

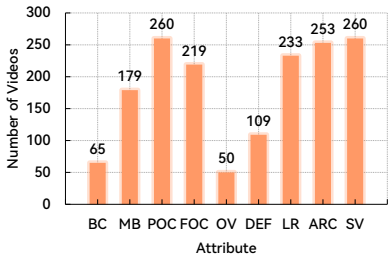
Att.	Definition
BC	The background has similar appearance as the target
MB	The target region is blurred due to target motion
POC	The target is partially occluded in the frame
FOC	The target is fully occluded in the frame
OV	The target completely leaves the video frame
DEF	The target is deformable during tracking
LR	The target BBox is smaller than 1000 pixels
ARC	The ratio of BBox aspect ratio is outside the range [0.5, 2]
SV	The ratio of BBox is outside the range [0.5, 2]

Table 4. Description of 9 attributes in MVTrack dataset.

Statistical Details. The MVTrack dataset contains 260 videos averaging around 900 frames each, as shown in Figure 7a. As illustrated in Figure 7b, a key challenge is occlusion, which often results from subject-object interactions that cause partial or complete occlusion. Consequently, tracking models need to manage occlusion to perform robustly and adeptly on this dataset.



(a) Frames distribution.



(b) Attributes distribution.

Figure 7. Distribution of sequences in each attribute and length in our MVTrack dataset.

10. Experiment Details

10.1. Training and Resource Analysis

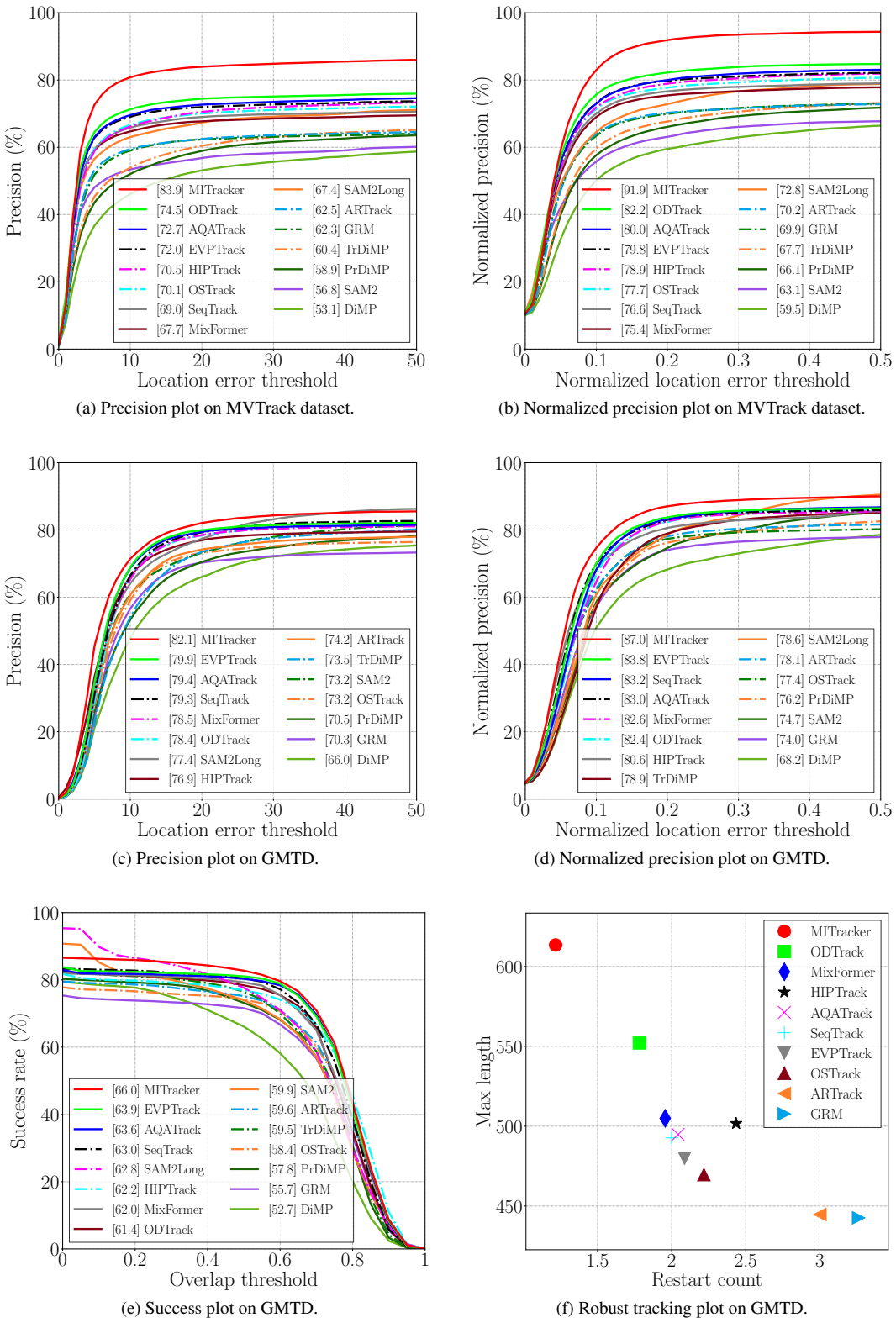
Training Details. We process the visual inputs by cropping the reference frame to 2 times the target's BBox size and resizing it to 182×182 pixels. The search frame is cropped at 4.5 times the target box area and resized to 364×364 pixels to expand the search region. During projection, we transform the camera intrinsic matrix C_K accordingly and add noise to the translation vector C_t to prevent overfitting in multi-view fusion.

Training consists of two stages. In the first stage, we optimize the view-specific encoder using AdamW with a learning rate of 1×10^{-5} and the rest of the model at 1×10^{-4}. We train for 50 epochs, sampling 10,000 image pairs per epoch with a batch size of 32. In the second stage, we fine-tune the encoder at 1×10^{-6} while keeping other components at 1×10^{-4}. We use the MVTrack dataset, sampling 2,500 multi-view image pairs per epoch for 40 epochs with a batch size of 4. AdamW is used throughout.

Computational Resource. We evaluate MITracker and the single-view model ODTrack under the same input (4 views) on an NVIDIA A100, as summarized in Table 5. Although multi-view fusion introduces additional computational overhead, it remains within an acceptable range.

Method	Parameters (M)	GRAM (MB)	FPS
ODTrack	92.12	365.82	18.78
MITracker	101.65	407.78	14.08

Table 5. Comparison of computational complexity and resource.



(a) Precision plot on MVTrack dataset.

(b) Normalized precision plot on MVTrack dataset.

(c) Precision plot on GMTD.

(d) Normalized precision plot on GMTD.

(e) Success plot on GMTD.

(f) Robust tracking plot on GMTD.

Figure 8. Comparative results across MVTrack and GMTD datasets, with rankings noted in the legends. Parts (a) and (c) sort methods by P with a 20-pixel threshold, parts (b) and (d) by P_{norm} with a 0.2 threshold, and part (e) by AUC.

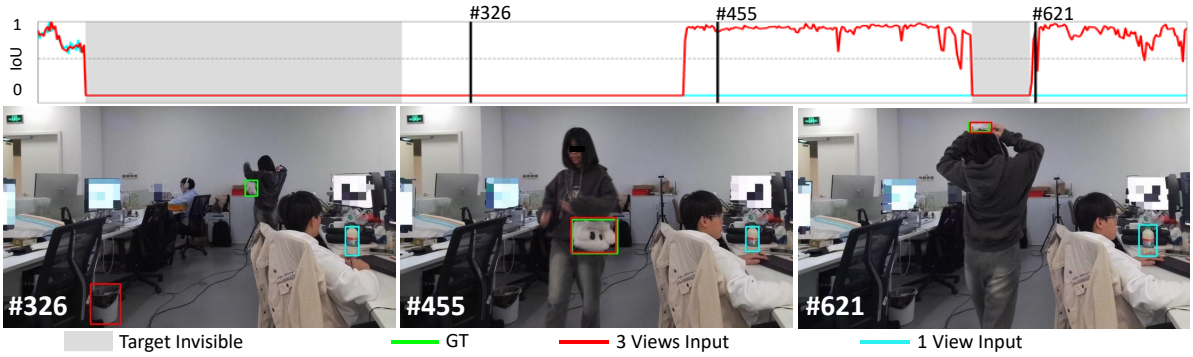


Figure 9. Qualitative comparison results on the impact of different numbers of input views. For a specific view, we compare the effects of using only that view versus including two additional overlapping views.

10.2. Comparison on Benchmark Details

In Figure 8, we provide further quantitative evaluations of the AUC, P, and P_{norm} across various threshold settings for both the MVTrack and GMTD datasets. In most settings, MITracker consistently outperforms other methods.

During zero-shot testing on the GMTD, SAM2 and SAM2Long perform better under lenient threshold conditions but lacks the ability to localize objects precisely. Furthermore, as shown in Figure 8f, MITracker sustains longer tracking durations with fewer reinitializations on this unseen dataset.

10.3. More Ablation Study

Impact of Input Views. To assess the importance of the number of views for tracking, we select a fixed camera from each scenario in the testing set. We then examine how model performance changes as we increase the number of additional cameras. The results in Table 6 highlight the benefits of adding more cameras.

Figure 9 illustrates the challenges faced by the single-view model: after a prolonged target disappearance, it mistracks a white bottle. In contrast, the multi-view model initially mistakes a white trash can for the target but quickly recovers and maintains stable tracking with the aid of additional views.

Input views	AUC(%)	P_{Norm}(%)	P(%)
1	62.27	84.71	73.92
2	63.97	87.07	76.30
3	67.97	91.50	80.73
3/4	68.65	92.37	81.55

Table 6. Ablation study for the impact of different numbers of input views on MVTrack dataset.

Impact of Multi-View Training. Our experiment shows that multi-view training improves single-view performance by exposing the model to richer spatial information, which enhances its ability to handle occlusion and reappearance. Table 7 compares results with MITracker SV trained under single-view settings, highlighting the advantages of multi-view training even for single-view scenarios.

Method	AUC(%)	P_{Norm}(%)	P(%)
MITracker SV	63.42	82.97	79.67
MITracker	65.96	87.05	82.07

Table 7. Zero-shot performance of single-view results on GMTD.

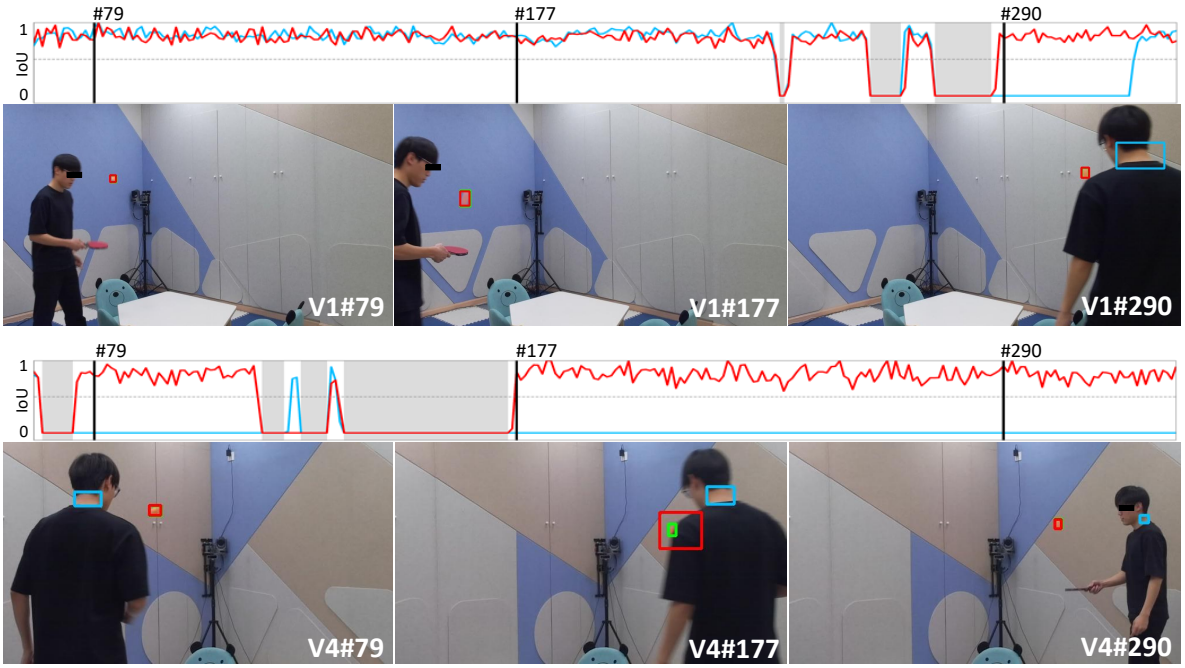
Impact of Temporal Token. The temporal token incorporates tracking information from previous frames, Table 8 highlights the improvements achieved through the temporal token.

Temporal Token	AUC(%)	P_{Norm}(%)	P(%)
	69.30	89.62	81.60
✓	71.13	91.87	83.95

Table 8. Ablation study for temporal token.

10.4. More Visualization Results

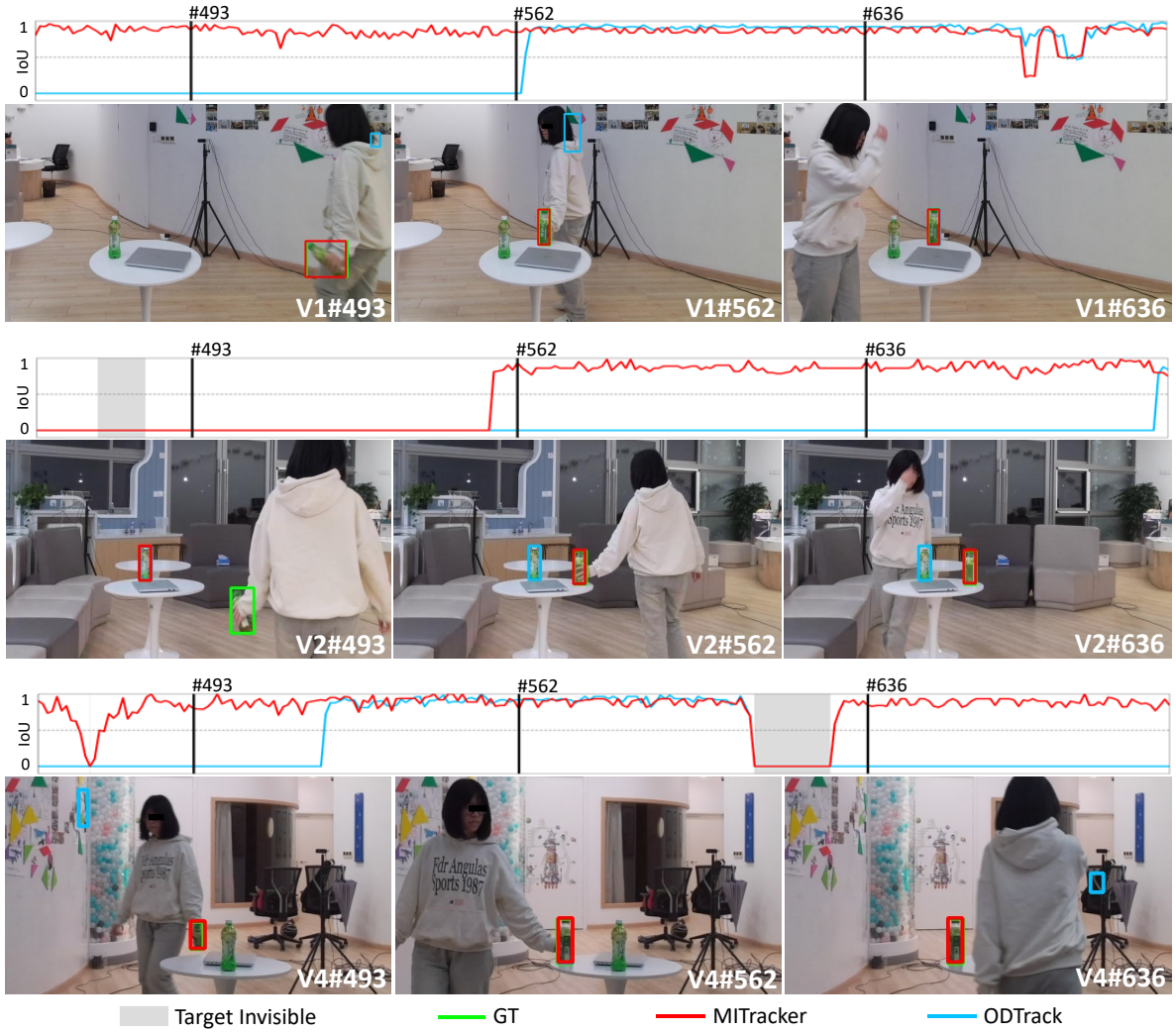
We provide additional visual comparison results as illustrated in Figure 10 and Figure 11 from the MVTrack dataset, and Figure 12 from the GMTD. MITracker exhibits enhanced re-tracking capabilities both in multi-view and single-view scenarios. Furthermore, multi-view information assists in correcting instances of mistracking. To facilitate better visualization, each frame is cropped to a fixed area. The IoU curves above further illustrate the tracking accuracy by comparing each method's predictions to the ground truth.



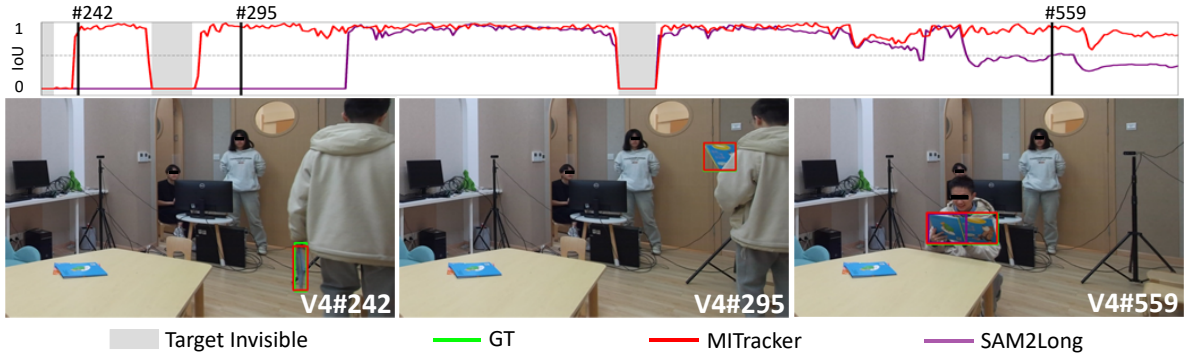
(a) Two views: *pingpong5-1* and *pingpong5-4*. ODTrack tends to lose track after extended periods of target disappearance, whereas MITracker demonstrates robust recovery capabilities.

(b) Two views: *umbrella2-1* and *umbrella2-2*. Under the interference of a similar object, ODTrack fails to re-track the correct target. In contrast, with the aid of multi-view assistance, MITracker can correct tracking errors from frame V1#415 to #521.

Figure 10. Qualitative comparison results on the MVTrack dataset using ODTrack.

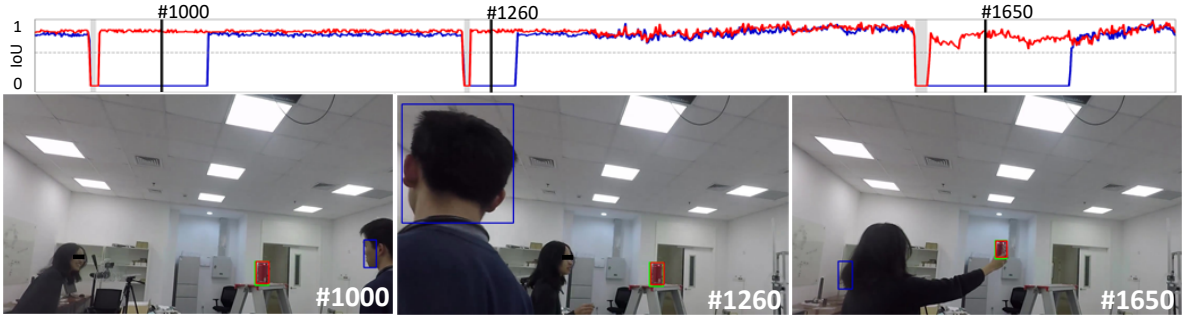


(a) Three views: *bottle3-1*, *bottle3-2* and *bottle3-4*. In V2 #493, MITracker momentarily mistracks a similar object as the target but successfully re-tracks the target by #562. In contrast, ODTrack struggles to recover once it mistracks.

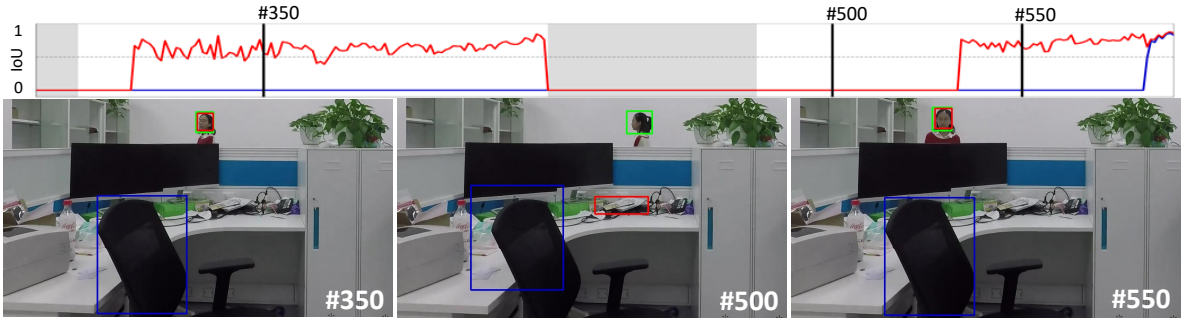


(b) Sequence: *book4-4*. SAM2Long completely loses the target following disappearances at frames #242 and #295. Upon re-tracking, it fails to adapt to target deformation, resulting in diminished IoU by frame #559.

Figure 11. Qualitative comparison results on the MVTrack dataset using ODTrack and SAM2Long.



(a) Sequence: *cola-2*. MITracker demonstrates faster re-tracking capabilities than EVPTrack upon target reappearance.



(b) Sequence: *manInOffice-2*. EVPTrack fails to correct after mistracking. In contrast, MITracker exhibits superior recovery capabilities, as demonstrated between frames #500 and #550.

Figure 12. Qualitative comparison results on the GMTD using EVPTrack.