Is Tracking really more challenging in First Person Egocentric Vision?

Supplementary Material

In this appendix, we present additional motivations, details, and results regarding our benchmark study.

We emphasize that the goal of this paper is not to develop new tracking methods for FPV but to enhance understanding of the task of object tracking in FPV. We aim to provide deeper insights that can better guide the future development of object tracking algorithms.

A. Details on the VISTA Benchmark

A.1. Background

Previous VOT and VOS benchmarking studies in egocentric vision have employed the following method to quantify the performance of a tracking algorithm. They considered an annotated video as a pair $\mathcal{V} = (\mathcal{F}, \mathcal{A})$ where $\mathcal{F} = \{\mathbf{F}_t\}_{t=0}^{T-1}$ and $\mathcal{A} = \{\mathbf{A}_t\}_{t=0}^{T-1}$ are, respectively, a sequence of T RGB frames $\mathbf{F}_t$, and a sequence of T annotations $\mathbf{A}_t$ in the form of bounding-boxes $\mathbf{b}_t$ [10, 49] or segmentation masks $\mathbf{M}_t$ [6]. Annotations are provided for all or a subset of the frames, in the latter case $\mathbf{A}_t = \emptyset$ for some t. The evaluation protocol used first initialized the tracker using the first frame $\mathbf{F}_0$ and its corresponding annotation $\mathbf{A}_0$. Then, the tracker was run on all subsequent frames $\mathbf{F}_t, t > 0$, producing a set of predictions $\mathcal{P} = \{\mathbf{P}_t\}_{t=1}^{T-1}$ represented as boxes $\widehat{\mathbf{b}}_t$ or masks $\widehat{\mathbf{M}}_t$. This protocol, adopted from popular VOT and VOS benchmarks, is defined as the one-pass evaluation (OPE) protocol in VOT [53, 54] and as semi-supervised evaluation in VOS [43]. To obtain a score expressing the quality of the behavior of the algorithm, the predictions were compared to the ground-truth annotations using a scoring function $\sigma\big(\{\mathbf{P}_t\}_{t=1}^{T-1}, \{\mathbf{A}_t\}_{t=1}^{T-1}\big)$. This process was repeated across multiple sequences, and the scores were averaged to produce a single metric that quantifies the algorithm's overall performance [6, 10, 49].

The benchmarking efforts employing this schema [6, 10, 49] compared the obtained scores with those achieved with the same protocol on established VOT and VOS benchmarks [13, 26, 43, 53, 55], highlighting a performance decline used to claim that FPV is more challenging than TPV. However, this comparison has several limitations. The overall scores come from different data domains, leading to inconsistencies due to differences in object categories and behaviors, sequence lengths, annotation rates, and dataset sizes. Additionally, training sets for training tracking models were drawn from mismatched data distributions. These factors can obscure the true impact of the FPV viewpoint and potentially mislead conclusions about VOTS algorithm performance in egocentric vision. It is worth mentioning that these issues may affect any benchmark dataset that differs significantly from established ones. In this paper, we specifically focus on egocentric FPV because it was often claimed to be particularly challenging.

A.2. Single Object Tracking

In this paper, we focus on tracking a single object per video. This choice to restrict the analysis to a single object is to obtain a more detailed examination of the key challenges and factors affecting FPV and TPV tracking. This approach ensures that the evaluation remains unaffected by the complexities introduced by multi-object interactions. Future work could explore multi-object tracking (MOT) evaluation approaches [34] to achieve a more comprehensive understanding of the impact of FPV and TPV on MOT algorithms. We believe that the insights provided in this study will be valuable for the development an benchmarking of such methods in FPV and TPV.

A.3. Online Evaluation and Initialization

In designing SOPE, we adhere to the OPE [54] and semi-supervised protocols [43], which process video frames sequentially in an online manner. This ensures fair comparison with previous benchmark studies [6, 10, 49] while also reflecting real-world scenarios where VOTS algorithms must operate in real-time, processing streaming video from wearable cameras for timely video understanding and user assistance.

For tracker initialization in SOPE, we follow again the OPE [54] and semi-supervised protocols [43], where the target is initialized in the first frame of the sequence. While user-provided initialization is less common in egocentric vision, prior work [10, 14, 37] has shown that visual trackers can be initialized by object detectors in the context of tracking-based downstream tasks. Thus, SOPE's standard initialization—using the target's first appearance and initial localization in the two views—remains relevant with respect to the real-world usage of a tracker.

A.4. Video Collection

The video sequences in the VISTA benchmark were selected from the EgoExo4D dataset [16], which is currently the largest resource for studying and developing human activity understanding algorithms from synchronized egocentric (FPV) and exocentric (TPV) point of views. It contains 1,422 hours of video featuring diverse activities such as sports, music, dance, and bike repair, collected from over 800 participants across 13 cities worldwide. Object categories relate to the activities performed in the videos and include kitchen tools, working tools, appliances, sport

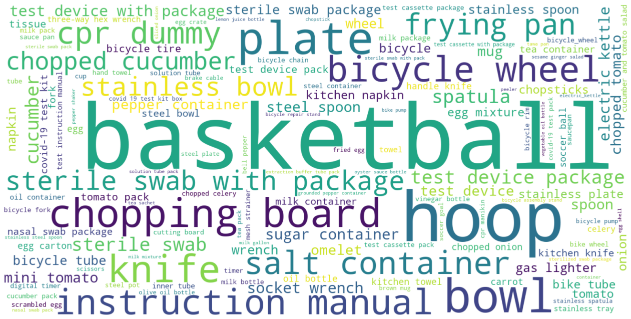

Figure 8. **Object categories represented in VISTA.** This wordcloud visualizes the categories and the frequency of the target objects available in our benchmarks's training and test sets.

equipment, kit parts, etc. The VISTA dataset consists of 6 human-object activities — bike repair, cooking, basketball, cardiopulmonary resuscitation, COVID testing, and soccer — featuring circa 285 distinct object categories (see Fig. 8). The videos are captured by 181 unique camera wearers in 53 different environments across 12 institutions. Ego-Exo4D's extensive coverage makes it a unique dataset, capturing a wide range of real-world scenarios with different individuals and various object types from both FPV and TPV perspectives. This diversity allowed us to curate a set of FPV and TPV tracking sequences that accurately reflect real-world application scenarios.

The FPV videos in Ego-Exo4D are recorded using Aria smart glasses [12], equipped with an 8 MP RGB camera capturing frames at a resolution of 1408 × 1408. For TPV perspectives, multiple stationary GoPro cameras are used, producing landscape videos with a resolution of 1920 × 1280. The placement and number of these exocentric cameras vary per scenario to ensure optimal coverage without obstructing participants' activities [16]. In each FPV-TPV pair, we select the TPV view that was originally annotated with object tracks in Ego-Exo4D, as it provides the clearest observation of the scene [16]. To optimize storage, we performed experiments by resizing FPV frames to 720 × 720 and TPV frames to 1280 × 720.

A.5. Bounding-box Annotations

The ground-truth axis-aligned bounding-box annotations $\{\mathbf{b}_t^{pov}\}_{t=0}^{T-1}$ for each FPV-TPV video in VISTA are derived by determining the minimum and maximum x, y coordinates of the positive pixels in the corresponding segmentation masks $\{\mathrm{M}_t^{pov}\}_{t=0}^{T-1}$.

A.6. Frame Attributes

The sequences have been annotated with 12 attributes that characterize motion and visual appearance changes affecting the target object. These attributes help analyze tracker performance under various conditions that may impact its behavior. Each attribute has been assigned on a per-frame basis to enable a more robust evaluation, following the approach in [24]. They have been selected from commonly used attributes in previous tracking benchmarks [13, 20, 24, 54] to ensure they capture scenarios relevant to both FPV and TPV videos. Following [13, 20, 24], the attributes were initially assigned using an automatic approach and later verified by our research team, consisting of two postdoctoral researchers and a full professor with expertise in visual tracking. Below, we describe the characteristics each attribute represents. The procedures outlined are applied independently to the FPV and TPV sequences.

- *Scale Variation (SV).* A scale variation in the object's appearance occurs if the ratio of the bounding-box area between the first and the current frame falls outside the range [0.5, 2] [13, 54].
- *Aspect Ratio Change (ARC).* An aspect ratio change occurs if the ratio of the bounding-box aspect ratio between the first and the current frame falls outside the range [0.5, 2] [13, 54].
- *Illumination Variation (IV).* Illumination variation occurs if the target's bounding-box is subject to significant light changes. The degree of illumination variation in each frame is measured by the change in average color between the first and the current target patch, following [20]. A threshold of 0.15 is used to determine illumination variation.
- *Distractors (DIS).* A frame contains distractors if it includes objects similar to the target, either from the same category or with a visually similar appearance. To identify distractors, we run a SAM2 instance [47] on each frame using a dense grid of point prompts to extract candidate object positions. Each candidate is then verified using DinoV2 [42] by computing the cosine similarity between its extracted features and those of the target crop from the first frame. A candidate is considered a distractor if its cosine similarity exceeds 0.5 and its bounding-box overlap with the ground-truth is below 0.5.
- *Motion Blur (MB).* Motion blur occurs when the target region appears blurred due to object or camera motion. Following [23, 24], we detect motion blur by computing the variance of the Laplacian on the target patch in the current frame. A threshold of 100 is used to determine the presence of motion blur.
- *Fast Motion (FM).* Fast motion is detected when the target bounding-box moves a distance greater than its own size between consecutive frames [13]. This attribute is computed by measuring the displacement of the bounding box during periods of target visibility.
- *Low Resolution (LR).* The target patch is considered low resolution if the area of the target bounding-box is smaller than 32^2 pixels [32].

- *Medium Resolution (MR).* The target patch is considered medium resolution if the area of the target bounding-box is between 32^2 and 96^2 pixels [32].
- *High Resolution (HR).* The target patch is considered high resolution if the area of the target bounding-box is larger than 96^2 pixels [32].

In addition to these standard attributes, we computed the following:

- *Static Object (STA) and Moving Object (MOV).* The object is considered static in the current frame if it remains in the same position relative to the previous frame. This is determined by computing the IoU between the bounding-box of the current and previous frame. If the IoU is above 0.5, the target object is labeled as static; otherwise, it is labeled as moving. This information is computed using only the TPV view, leveraging the synchronization of frames and annotations to assign the corresponding label to the associated FPV frame. Since the TPV camera is stationary, any change in annotation overlap during target visibility periods is due to the motion of the object.
- *Hand-Object Interaction (HOI).* The target object is considered to be in interaction with a person's hands. Following [10, 14], we compute this attribute by first running a hand-object interaction detector [48]. To determine whether the target object is being interacted with, we check if its bounding-box overlaps with the detected object bounding boxes by more than 0.5 IoU and if an interaction state is detected for at least two consecutive annotations (equivalent to a period of 1 second). Once an interaction label is assigned, it remains active for all subsequent frames until two consecutive overlaps fall below 0.5 with no interaction state detected [14]. This information is computed using only the FPV view becuase the egocentric viewpoint enables a closer view of the interaction between hands and objects [48]. We leverage the synchronization of frames and annotations to assign the corresponding label to the associated TPV frame.

A.7. Metrics

As shown in Eq. 1, we compute the mean signed difference for each sequence, weight it by the annotation length, and then average it across the sum of all the annotation lengths [24]. We applied this weighting because we observed that methods tend to perform better on short FPV videos. Without weighting, a high score from a short FPV sequence carries the same influence as a low score from a long FPV sequence, which can mask the overall lower tracking accuracy of longer videos. Conversely, algorithms tend to perform better on long TPV videos. In this case, a long TPV sequence with limited object motion and a high score would carry the same weight as a short TPV sequence with object motion and poor performance, potentially overshadowing the true performance trend. In Tab. 3, we report the difference in using or not using the weighting on the AUC, NPS, and GSR metrics. Tab. 5 shows the impact of not having the weighting as used by the standard VOS benchmark evaluation [43, 55].

The same weighting strategy described before is applied to the standard metrics reported in all tables and figures of this paper, specifically AUC-pov, NPS-pov, or GSR-pov. This is represented by the following equation:

$$s_{\sigma}^{pov} = \frac{1}{\omega_0 + \cdots + \omega_{N-1}} \sum_{i=0}^{N-1} s_{\sigma,i}^{pov} \cdot \omega_i, \omega_i = |\mathcal{A}_i^{pov}| \quad (2)$$

where $s_{\sigma,i}^{pov}$ represents the AUC, NPS, or GSR score for an individual sequence.

We compute the sequence-wise scores $s_{\sigma,i}^{pov}$ for bounding-box trackers by calculating the AUC, NPS, and GSR based on the IoU between the predicted bounding-boxes $\{\widehat{\mathbf{b}}_t^{pov}\}_{t=1}^{T-1}$ and the ground-truth bounding-boxes $\{\mathbf{b}_t^{pov}\}_{t=1}^{T-1}$. For trackers that output segmentation masks, we compute the AUC, NPS, and GSR based on the IoU between the predicted segmentation masks $\{\widehat{\mathbf{M}}_t^{pov}\}_{t=1}^{T-1}$ and the ground-truth masks $\{\mathbf{M}_t^{pov}\}_{t=1}^{T-1}$. This approach ensures a fair evaluation of the tracker's predictions by comparing them to the ground-truth target state representation that the tracker was optimized for.

In addition to the previously mentioned metric, Table 5 reports the scores $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$, which are commonly used for VOS evaluation. These scores were computed as originally described in [43, 55]. As with the other metrics, we calculate the mean signed differences $\Delta_{\mathcal{J}\&\mathcal{F}}, \Delta_{\mathcal{J}}, \Delta_{\mathcal{F}}$ to quantify the performance differences between FPV and TPV based on these metrics. For these segmentation-oriented metrics, we convert the bounding-boxes predicted by box-based trackers into segmentation masks by filling the rectangular area within the bounding-box with positive pixels [27, 28].

B. Details on the Evaluated Methods

For SAM2-based instances [47] (SAM2-B, SAM2-M, SAMURAI, DAM4SAM), we used the SAM 2++ Hiera Large instance. SAM2-B is a variant of SAM2 initialized with a bounding box and producing bounding-box outputs. For SAMURAI, we follow the running setup from the original paper [58], initializing it with a bounding-box and retrieving output bounding-boxes from it.

In the following, we provide details about the viewpoint-optimized tracking baselines mentioned in Sec. 4 of the main paper. For STARK-T-FPV, we train the STARK-ST50 instance [56] starting from random weights on the FPV sequences of the $\mathcal{D}_{\text{TRAIN}}$ in the VISTA benchmark. The training consists of 100 epochs in stage 1 and 10 epochs in stage 2. Apart from the number of epochs, all other hyperparameters are kept fixed as originally proposed [56].

Table 3. **Effect of score weighting by sequence length.** This table shows how performance difference scores change when each sequence score is weighted by its annotation length. We apply this weighting because trackers behave differently across the two viewpoints, and unweighted scores from short or long videos can distort the true average performance.

	Tracker	Weight	AUC			NPS			GSR		
			FPV	TPV	Δ_{AUC}	FPV	TPV	Δ_{NPS}	FPV	TPV	Δ_{GSR}
⬠	STARK-F-FPV	✓	51.9	44.2	7.7	54.0	45.6	8.4	11.3	28.7	-17.4
		✗	54.0	43.7	10.3	56.0	45.1	10.9	21.3	33.7	-12.4
⬠	XMem-F-FPV	✓	56.3	52.6	3.7	60.8	57.6	3.2	29.8	39.4	-9.6
		✗	58.7	51.9	6.8	63.6	57.0	6.6	42.9	45.6	-2.7
❋	DAM4SAM	✓	61.7	65.8	-4.1	66.2	71.1	-4.9	35.2	52.9	-17.7
		✗	64.4	63.5	0.9	69.6	69.0	0.6	47.3	57.6	-10.3
✻	SAMURAI	✓	56.2	63.3	-7.1	58.3	65.6	-7.3	31.8	48.2	-16.4
		✗	60.6	63.1	-2.5	63.1	65.3	-2.2	43.9	53.5	-9.6
⬠	STARK-F-TPV	✓	42.2	51.7	-9.5	43.3	53.9	-10.6	7.9	33.0	-25.1
		✗	44.5	50.6	-6.1	45.8	52.4	-6.6	16.7	37.7	-21.0
⬠	XMem-F-TPV	✓	40.2	52.6	-12.4	45.7	58.5	-12.8	18.9	39.7	-20.8
		✗	45.0	53.0	-8.0	50.8	59.0	-8.2	30.1	46.4	-16.3

For STARK-T-TPV, we use the same training configuration, with the only change being the training set, which consists of the TPV sequences from $\mathcal{D}_{\text{TRAIN}}$. For STARK-T-FPV,TPV (row 4 of Tab. 2), we use the same training configuration as described previously, with the only change being the training set, which includes both FPV and TPV sequences from $\mathcal{D}_{\text{TRAIN}}$. For STARK-F-FPV, STARK-F-TPV, and STARK-F-FPV,TPV, we follow the same approach as before, but start with pretrained model weights obtained after training for generic object tracking on the TrackingNet [40], LaSOT [13], GOT-10k [20], and COCO [32] datasets. The original code repository was used to implement all of these procedures.[2]

For the XMem baseline, we follow a similar approach. For XMem-T-FPV, we the ResNet50-based instance [3] starting from weights pretrained on static images (stage 0) on the FPV sequences of $\mathcal{D}_{\text{TRAIN}}$. The training kept all hyperparameters fixed as originally proposed [3]. For XMem-T-TPV, we use the same training configuration, with the only change being the training set, which consists of the TPV sequences from $\mathcal{D}_{\text{TRAIN}}$. For XMem-T-FPV,TPV (row of Tab. 2), we use the same training configuration as described previously, with the only change being the training set, which includes both FPV and TPV sequences from $\mathcal{D}_{\text{TRAIN}}$. For XMem-F-FPV, XMem-F-TPV, and XMem-F-FPV,TPV, we follow the same approach as before, but start with pretrained model weights obtained after training on generic object VOS using the DAVIS [43], and YouTube-VOS [55] datasets (stage 3). The original code repository was used to implement all of these procedures.[3]

All the code used for this study was implemented in Python and run on a machine with an Intel Xeon E5-2690 v4 @ 2.60 GHz CPU, 320 GB of RAM, and 6 NVIDIA TITAN V GPUs.

C. Details and Additional Results

In all Figures and Tables in Sec. 5 of the main paper, unless stated otherwise, results are based on the 544 pairs in $\mathcal{D}_{\text{TEST}}$ under the long-term object tracking setting.

Frame-based attribute evaluation. For experiments involving frame-based attributes, scores are computed using only frames labeled with the respective attribute. In these cases, each sequence is weighted based on the number of annotated frames containing the attribute of interest.

Long-term tracking scores. For improved readability, the full version of the scores shown in brackets in Fig. 3 is provided in Tab. 4. Refer to Sec. 5 of the main paper for a detailed discussion.

Details on experiments on the field's of view impact. To generate the results shown in Fig. 5 of the main paper, we categorized each annotation based on its distance from the frame center into four regions: (1) within 25% of the frame width from the center, (2) between 25% and 50% of the frame width, (3) between 50% and 75% of the frame width,

[2] https://github.com/researchmm/Stark

[3] https://github.com/hkchengrex/XMem

Table 4. **Long-term object tracking performance across FPV and TPV.** For a better readibility, this table reports the score presented in Figure 3 (a). Light blue represents FPV bounding-box trackers; dark blue represents FPV segmentation trackers; light red represents TPV bounding-box trackers; dark red represents TPV segmentation trackers; light green represents generic bounding-box trackers; dark green represents generic segmentation trackers. Trackers are ordered in descending order by Δ_{AUC}.

	Tracker		AUC			NPS			GSR	
		FPV	TPV	Δ_{AUC}	FPV	TPV	Δ_{NPS}	FPV	TPV	Δ_{GSR}
⬡	STARK-T-FPV	49.8	38.4	11.4	51.8	39.4	12.4	10.9	24.8	-13.9
⬠	STARK-F-FPV	51.9	44.2	7.7	54.0	45.6	8.4	11.3	28.7	-17.4
⬡	XMem-T-FPV	56.7	49.2	7.5	61.0	54.5	6.5	30.9	37.7	-6.8
⬠	XMem-F-FPV	56.3	52.6	3.7	60.8	57.6	3.2	29.8	39.4	-9.6
⋈	EgoSTARK	45.0	44.7	0.3	46.4	46.5	-0.1	10.0	29.7	-19.7
⋇	DAM4SAM	61.7	65.8	-4.1	66.2	71.1	-4.9	35.2	52.9	-17.7
⊗	TaMOs	34.5	39.0	-4.5	36.2	39.5	-3.3	10.6	26.8	-16.2
×	AOT	37.5	42.6	-5.1	38.7	43.4	-4.7	11.4	30.7	-19.3
◇	OSTrack	43.7	49.2	-5.5	45.3	51.0	-5.7	10.2	31.3	-21.1
★	KeepTrack	34.5	40.5	-6.0	36.9	42.8	-5.9	8.5	29.0	-20.5
∗	SAMURAI	56.2	63.3	-7.1	58.3	65.6	-7.3	31.8	48.2	-16.4
⋈	STARK	35.5	42.8	-7.3	36.6	44.6	-8.0	8.1	28.1	-20.0
◀	UNICORN-M	23.9	32.5	-8.6	28.4	40.4	-12.0	5.8	19.2	-13.4
⊞	XMem	37.1	45.9	-8.8	40.3	49.4	-9.1	17.8	35.2	-17.4
⬠	STARK-F-TPV	42.2	51.7	-9.5	43.3	53.9	-10.6	7.9	33.0	-25.1
⬡	STARK-T-TPV	36.8	48.0	-11.2	38.0	50.0	-12.0	6.9	29.9	-23.0
⬠	XMem-F-TPV	40.2	52.6	-12.4	45.7	58.5	-12.8	18.9	39.7	-20.8
▶	SAM2-M	45.7	58.8	-13.1	49.1	64.1	-15.0	24.0	47.9	-23.9
○	SeqTrack	36.9	50.3	-13.4	39.0	52.8	-13.8	8.1	33.7	-25.6
●	ARTrackV2	32.4	46.0	-13.6	32.6	47.7	-15.1	7.5	33.9	-26.4
▶	SAM2-B	45.8	59.9	-14.1	47.9	62.8	-14.9	22.7	43.3	-20.6
◀	UNICORN-B	27.0	41.3	-14.3	28.1	42.1	-14.0	6.7	26.1	-19.4
◁	Cutie	30.6	46.6	-16.0	33.4	50.9	-17.5	15.6	36.4	-20.8
⬡	XMem-T-TPV	33.6	51.5	-17.9	37.8	56.9	-19.1	14.6	39.6	-25.0

and (4) beyond 75% of the frame width. The position of each annotation was determined using the coordinates of its barycenter. This clustering process was applied separately to FPV and TPV. To compute the scores for each cluster, we followed the same procedure used for frame attribute-based evaluation. For FPV, each cluster contains 11% (25%), 33% (25-50%), 30% (50-75%), 26% (75-100%) of the total annotated frames. For TPV, each cluster contains 14% (25%), 31% (25-50%), 26% (50-75%), 29% (75-100%) of the total annotated frames.

VOS-based evaluation results. Tab. 5 presents the performance scores of the selected trackers using the standard semi-supervised evaluation protocol, measured with the $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$ metrics [43]. The $\mathcal{J}$ metric quantifies the average overlap between the predicted and ground-truth segmentation masks (similar to AUC), while $\mathcal{F}$ assesses the quality of segmentation boundaries. The $\mathcal{J}\&\mathcal{F}$ metric is the average of the two. To analyze viewpoint-dependent performance differences, we compute the signed difference for these metrics.

The results confirm the conclusions drawn from Fig. 3 and Tab. 4. Generic object trackers exhibit a more significant performance drop in FPV compared to TPV. Additionally, the bias introduced by viewpoint-optimized trackers is reflected also in these metrics. It is important to note that standard VOS evaluation does not weight sequence scores by sequence or annotation length. In this approach, all sequences contribute equally, regardless of their duration or annotation frequency, even though these factor can influence tracking performance scoring. As a result, this evaluation method fails to accurately capture viewpoint bias, as performance measurements become skewed toward short, high-scoring sequences, masking the true impact of viewpoint difference when trackers behave differently in the two views.

Qualitative Results. Fig. 9, 10, 11, and 12 present qualitative examples of the most accurate tracker, DAM4SAM [51], on selected FPV and TPV sequences from the VISTA

Table 5. **Object tracking performance across FPV and TPV with standard VOS metrics.** This Table reports $\mathcal{J}$ & $\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$ generally used in VOS evaluation [43, 55]. Similar conclusions to what reported for Fig. 3 can be made for these results. The computation of these metrics does not take into account the length of the sequence, and this can overshadow the real average FPV and TPV performance. Light blue represents FPV bounding-box trackers; dark blue represents FPV segmentation trackers; light red represents TPV bounding-box trackers; dark red represents TPV segmentation trackers; light green represents generic bounding-box trackers; dark green represents generic segmentation trackers. Trackers are ordered in descending order by $\Delta_{\mathcal{J}\&\mathcal{F}}$.

	Tracker		$\mathcal{J}$ & $\mathcal{F}$			$\mathcal{J}$			$\mathcal{F}$	
		FPV	TPV	$\Delta_{\mathcal{J}\&\mathcal{F}}$	FPV	TPV	$\Delta_{\mathcal{J}}$	FPV	TPV	$\Delta_{\mathcal{F}}$
⬡	XMem-T-FPV	64.2	57.3	6.9	59.0	49.3	9.7	69.4	65.2	4.2
⬠	XMem-F-FPV	63.9	59.3	4.6	58.7	51.2	7.5	69.1	67.4	1.7
⬡	STARK-T-FPV	35.8	31.5	4.3	34.3	25.2	9.1	37.3	37.7	-0.4
⬠	STARK-F-FPV	37.0	34.8	2.2	35.4	27.7	7.7	38.7	41.9	-3.2
⋇	DAM4SAM	69.4	70.1	-0.7	64.8	63.5	1.3	74.0	76.8	-2.8
⋈	EgoSTARK	31.4	33.9	-2.5	30.2	27.4	2.8	32.6	40.5	-7.9
×	AOT	42.9	46.9	-4.0	38.8	39.6	-0.8	47.1	54.2	-7.1
⊗	TaMOs	29.7	34.2	-4.5	27.8	25.8	2.0	31.6	42.7	-11.1
∗	SAMURAI	63.8	69.0	-5.2	59.2	60.9	-1.7	68.4	77.1	-8.7
◇	OSTrack	31.9	37.1	-5.2	30.6	30.0	0.6	33.2	44.2	-11.0
⊞	XMem	43.4	49.5	-6.1	40.1	43.6	-3.5	46.7	55.4	-8.7
⋈	STARK	27.7	34.1	-6.4	26.4	27.0	-0.6	29.0	41.2	-12.2
◀	UNICORN-M	32.8	41.0	-8.2	28.1	32.7	-4.6	37.6	49.2	-11.6
★	KeepTrack	28.2	36.6	-8.4	26.8	29.2	-2.4	29.7	44.1	-14.4
⬠	STARK-F-TPV	30.8	40.2	-9.4	29.5	31.8	-2.3	32.2	48.7	-16.5
◀	UNICORN-B	25.5	35.1	-9.6	23.4	27.7	-4.3	27.7	42.5	-14.8
◁	Cutie	41.0	50.6	-9.6	37.8	44.6	-6.8	44.3	56.6	-12.3
○	SeqTrack	27.1	37.4	-10.3	25.5	29.6	-4.1	28.6	45.2	-16.6
▶	SAM2-M	54.9	65.3	-10.4	51.4	58.5	-7.1	58.4	72.1	-13.7
⬡	STARK-T-TPV	28.1	38.8	-10.7	26.8	30.7	-3.9	29.4	46.9	-17.5
●	ARTrackV2	23.8	35.0	-11.2	23.1	28.2	-5.1	24.5	41.9	-17.4
⬠	XMem-F-TPV	49.5	60.7	-11.2	44.2	52.4	-8.2	54.7	69.0	-14.3
▶	SAM2-B	36.9	51.1	-14.2	34.9	41.9	-7.0	38.8	60.3	-21.5
⬡	XMem-T-TPV	44.0	59.3	-15.3	38.7	51.5	-12.8	49.3	67.1	-17.8

test set. Each figure displays the predicted target segmentations alongside the sequence-wise AUC-FPV, AUC-TPV, and their signed difference.

D. Limitations

This study did not control the placement of TPV cameras. Although we selected a TPV view that provides the best visualization of the scene and activity [16], future work could compare FPV with multiple TPV positions to assess their impact on tracking performance and explore how different TPV configurations relate to FPV.

This study did not assess the impact of FPV and TPV tracking on downstream tasks. Our focus was to evaluate performance differences between FPV and TPV in the VOTS task [16]. While prior work [10, 37] has shown a connection between tracking behavior and downstream task accuracy, we specifically examined object tracking performance. Future research could explore how the conclusions drawn from VISTA relate to the performance of higher-level vision modules that rely on FPV or TPV object tracking.

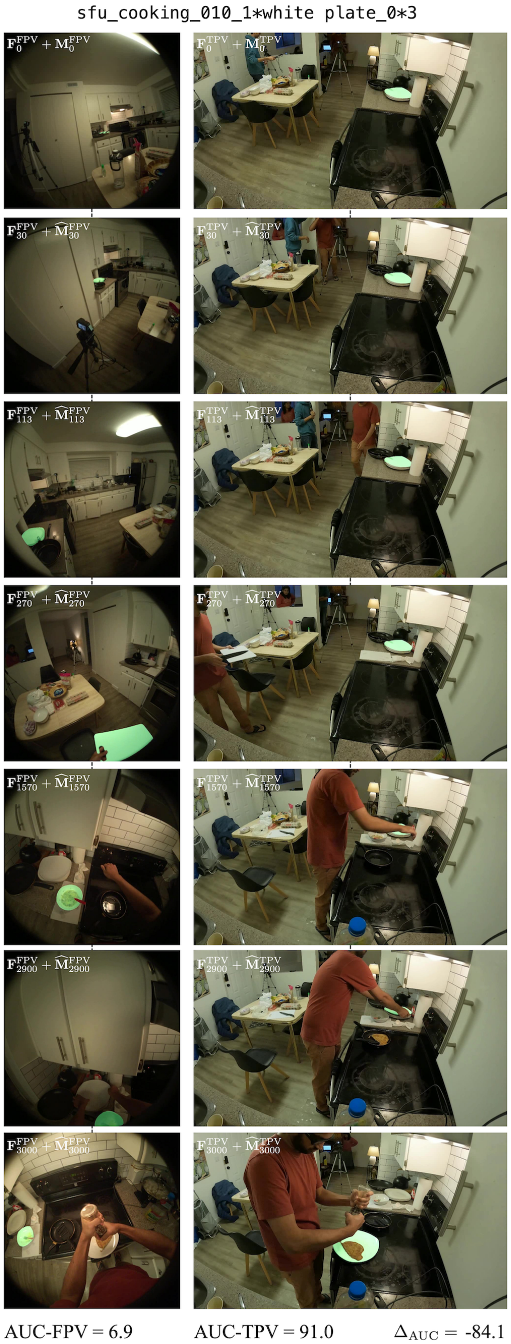
sfu_cooking_010_1*white plate_0*3
$\mathbf{F}_0^{\text{FPV}} + \mathbf{M}_0^{\text{FPV}}$
$\mathbf{F}_0^{\text{TPV}} + \mathbf{M}_0^{\text{TPV}}$
$\mathbf{F}_{30}^{\text{FPV}} + \widehat{\mathbf{M}}_{30}^{\text{FPV}}$
$\mathbf{F}_{30}^{\text{TPV}} + \widehat{\mathbf{M}}_{30}^{\text{TPV}}$
$\mathbf{F}_{113}^{\text{FPV}} + \widehat{\mathbf{M}}_{113}^{\text{FPV}}$
$\mathbf{F}_{113}^{\text{TPV}} + \widehat{\mathbf{M}}_{113}^{\text{TPV}}$
$\mathbf{F}_{270}^{\text{FPV}} + \widehat{\mathbf{M}}_{270}^{\text{FPV}}$
$\mathbf{F}_{270}^{\text{TPV}} + \widehat{\mathbf{M}}_{270}^{\text{TPV}}$
$\mathbf{F}_{1570}^{\text{FPV}} + \widehat{\mathbf{M}}_{1570}^{\text{FPV}}$
$\mathbf{F}_{1570}^{\text{TPV}} + \widehat{\mathbf{M}}_{1570}^{\text{TPV}}$
$\mathbf{F}_{2900}^{\text{FPV}} + \widehat{\mathbf{M}}_{2900}^{\text{FPV}}$
$\mathbf{F}_{2900}^{\text{TPV}} + \widehat{\mathbf{M}}_{2900}^{\text{TPV}}$
$\mathbf{F}_{3000}^{\text{FPV}} + \widehat{\mathbf{M}}_{3000}^{\text{FPV}}$
$\mathbf{F}_{3000}^{\text{TPV}} + \widehat{\mathbf{M}}_{3000}^{\text{TPV}}$
AUC-FPV = 6.9
AUC-TPV = 91.0
Δ_{AUC} = -84.1

Figure 9. **Qualitative example # 1.** Here, we illustrate the behavior of DAM4SAM [51] on a sequence from VISTA's evaluation set $\mathcal{D}_{\text{TEST}}$. The frames $\mathbf{F}_t^{pov}$ are overlaid with the predicted segmentation masks $\widehat{\mathbf{M}}_t^{pov}$ (shown in light green). Below the frames, we report the sequence-wise AUC-FPV, AUC-TPV, and Δ_{AUC}. In this example, the tracker loses the target early in the FPV sequence, whereas it remains stable in TPV despite object displacement. This discrepancy is reflected in the mean signed difference, which is highly negative.

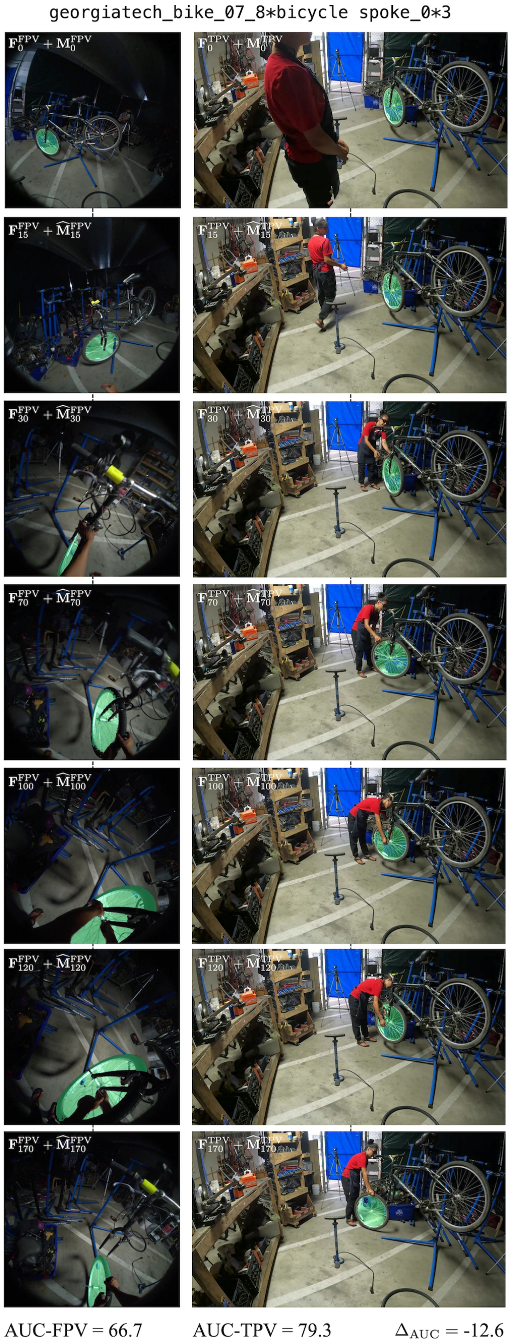
georgiatech_bike_07_8*bicycle spoke_0*3
$\mathbf{F}_0^{\text{FPV}} + \mathbf{M}_0^{\text{FPV}}$
$\mathbf{F}_0^{\text{TPV}} + \mathbf{M}_0^{\text{TPV}}$
$\mathbf{F}_{15}^{\text{FPV}} + \widehat{\mathbf{M}}_{15}^{\text{FPV}}$
$\mathbf{F}_{15}^{\text{TPV}} + \widehat{\mathbf{M}}_{15}^{\text{TPV}}$
$\mathbf{F}_{30}^{\text{FPV}} + \widehat{\mathbf{M}}_{30}^{\text{FPV}}$
$\mathbf{F}_{30}^{\text{TPV}} + \widehat{\mathbf{M}}_{30}^{\text{TPV}}$
$\mathbf{F}_{70}^{\text{FPV}} + \widehat{\mathbf{M}}_{70}^{\text{FPV}}$
$\mathbf{F}_{70}^{\text{TPV}} + \widehat{\mathbf{M}}_{70}^{\text{TPV}}$
$\mathbf{F}_{100}^{\text{FPV}} + \widehat{\mathbf{M}}_{100}^{\text{FPV}}$
$\mathbf{F}_{100}^{\text{TPV}} + \widehat{\mathbf{M}}_{100}^{\text{TPV}}$
$\mathbf{F}_{120}^{\text{FPV}} + \widehat{\mathbf{M}}_{120}^{\text{FPV}}$
$\mathbf{F}_{120}^{\text{TPV}} + \widehat{\mathbf{M}}_{120}^{\text{TPV}}$
$\mathbf{F}_{170}^{\text{FPV}} + \widehat{\mathbf{M}}_{170}^{\text{FPV}}$
$\mathbf{F}_{170}^{\text{TPV}} + \widehat{\mathbf{M}}_{170}^{\text{TPV}}$
AUC-FPV = 66.7
AUC-TPV = 79.3
Δ_{AUC} = -12.6

Figure 10. **Qualitative example # 2.** Here, we illustrate the behavior of DAM4SAM [51] on a sequence from VISTA's evaluation set $\mathcal{D}_{\text{TEST}}$. The frames $\mathbf{F}_t^{pov}$ are overlaid with the predicted segmentation masks $\widehat{\mathbf{M}}_t^{pov}$ (shown in light green). Below the frames, we report the sequence-wise AUC-FPV, AUC-TPV, and Δ_{AUC}. In this example, the tracker remains relatively stable in both FPV and TPV. However, the high variability of the target appearance in FPV makes segmentation prediction more challenging. This difference is reflected in the mean signed difference, indicating better performance in TPV.

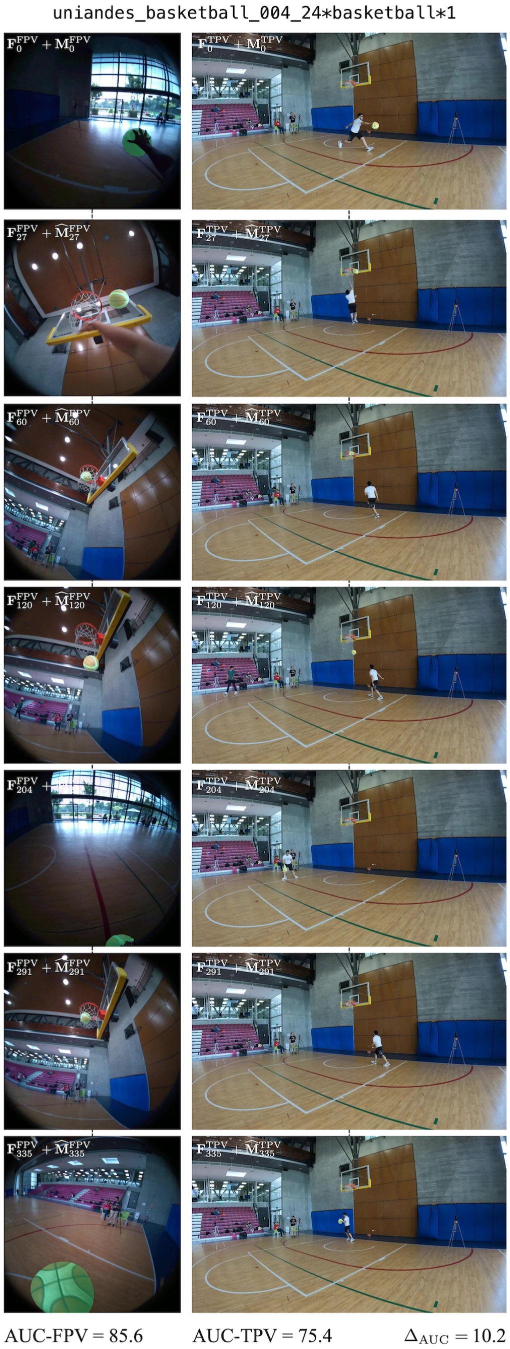
uniandes_basketball_004_24*basketball*1
$\mathbf{F}_0^{\text{FPV}} + \mathbf{M}_0^{\text{FPV}}$
$\mathbf{F}_0^{\text{TPV}} + \mathbf{M}_0^{\text{TPV}}$
$\mathbf{F}_{27}^{\text{FPV}} + \widehat{\mathbf{M}}_{27}^{\text{FPV}}$
$\mathbf{F}_{27}^{\text{TPV}} + \widehat{\mathbf{M}}_{27}^{\text{TPV}}$
$\mathbf{F}_{60}^{\text{FPV}} + \widehat{\mathbf{M}}_{60}^{\text{FPV}}$
$\mathbf{F}_{60}^{\text{TPV}} + \widehat{\mathbf{M}}_{60}^{\text{TPV}}$
$\mathbf{F}_{120}^{\text{FPV}} + \widehat{\mathbf{M}}_{120}^{\text{FPV}}$
$\mathbf{F}_{120}^{\text{TPV}} + \widehat{\mathbf{M}}_{120}^{\text{TPV}}$
$\mathbf{F}_{204}^{\text{FPV}}$
$\mathbf{F}_{204}^{\text{TPV}} + \widehat{\mathbf{M}}_{204}^{\text{TPV}}$
$\mathbf{F}_{291}^{\text{FPV}} + \widehat{\mathbf{M}}_{291}^{\text{FPV}}$
$\mathbf{F}_{291}^{\text{TPV}} + \widehat{\mathbf{M}}_{291}^{\text{TPV}}$
$\mathbf{F}_{335}^{\text{FPV}} + \widehat{\mathbf{M}}_{335}^{\text{FPV}}$
$\mathbf{F}_{335}^{\text{TPV}} + \widehat{\mathbf{M}}_{335}^{\text{TPV}}$
AUC-FPV = 85.6
AUC-TPV = 75.4
$\Delta_{\text{AUC}} = 10.2$

Figure 11. **Qualitative example # 3.** Here, we illustrate the behavior of DAM4SAM [51] on a sequence from VISTA's evaluation set $\mathcal{D}_{\text{TEST}}$. The frames $\mathbf{F}_t^{pov}$ are overlaid with the predicted segmentation masks $\widehat{\mathbf{M}}_t^{pov}$ (shown in light green). Below the frames, we report the sequence-wise AUC-FPV, AUC-TPV, and Δ_{AUC}. In this example, the tracker remains relatively stable in both FPV and TPV. However, the lower resolution of the target in TPV makes segmentation prediction more challenging. This difference is reflected in the mean signed difference, indicating better performance in FPV.

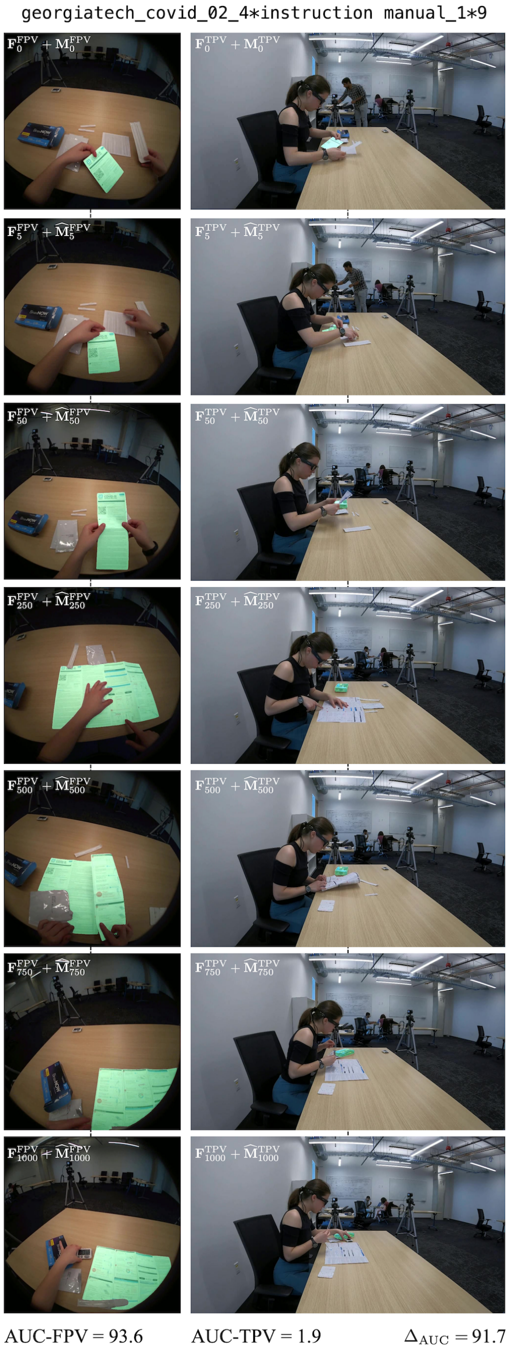
georgiatech_covid_02_4*instruction manual_1*9
$\mathbf{F}_0^{\text{FPV}} + \mathbf{M}_0^{\text{FPV}}$
$\mathbf{F}_0^{\text{TPV}} + \mathbf{M}_0^{\text{TPV}}$
$\mathbf{F}_5^{\text{FPV}} + \widehat{\mathbf{M}}_5^{\text{FPV}}$
$\mathbf{F}_5^{\text{TPV}} + \widehat{\mathbf{M}}_5^{\text{TPV}}$
$\mathbf{F}_{50}^{\text{FPV}} + \widehat{\mathbf{M}}_{50}^{\text{FPV}}$
$\mathbf{F}_{50}^{\text{TPV}} + \widehat{\mathbf{M}}_{50}^{\text{TPV}}$
$\mathbf{F}_{250}^{\text{FPV}} + \widehat{\mathbf{M}}_{250}^{\text{FPV}}$
$\mathbf{F}_{250}^{\text{TPV}} + \widehat{\mathbf{M}}_{250}^{\text{TPV}}$
$\mathbf{F}_{500}^{\text{FPV}} + \widehat{\mathbf{M}}_{500}^{\text{FPV}}$
$\mathbf{F}_{500}^{\text{TPV}} + \widehat{\mathbf{M}}_{500}^{\text{TPV}}$
$\mathbf{F}_{750}^{\text{FPV}} + \widehat{\mathbf{M}}_{750}^{\text{FPV}}$
$\mathbf{F}_{750}^{\text{TPV}} + \widehat{\mathbf{M}}_{750}^{\text{TPV}}$
$\mathbf{F}_{1000}^{\text{FPV}} + \widehat{\mathbf{M}}_{1000}^{\text{FPV}}$
$\mathbf{F}_{1000}^{\text{TPV}} + \widehat{\mathbf{M}}_{1000}^{\text{TPV}}$
AUC-FPV = 93.6
AUC-TPV = 1.9
$\Delta_{\text{AUC}} = 91.7$

Figure 12. **Qualitative example # 4.** Here, we illustrate the behavior of DAM4SAM [51] on a sequence from VISTA's evaluation set $\mathcal{D}_{\text{TEST}}$. The frames $\mathbf{F}_t^{pov}$ are overlaid with the predicted segmentation masks $\widehat{\mathbf{M}}_t^{pov}$ (shown in light green). Below the frames, we report the sequence-wise AUC-FPV, AUC-TPV, and Δ_{AUC}. In this example, the tracker loses the target early in the TPV sequence, whereas it remains stable in FPV despite object transformation. This discrepancy is reflected in the mean signed difference, which is highly positive.