A. Motivations And Details Behind TREK-150

In this section, we provide more motivations and details behind the construction of the TREK-150 benchmark dataset.

First of all, we remark that TREK-150 has been designed for the *evaluation* of visual tracking algorithms in FPV regardless of their methodology. Indeed, this paper does not aim to provide a large-scale dataset to improve the performance of deep learning based trackers. Instead, its goal is to assess the impact of the first-person viewpoint on current trackers and, to the best of our knowledge, this analysis was never done before. Hence, as a *first step* towards providing an answer to such a point (which is also highlighted in the title of the paper), we focused on benchmarking the tracking progress made by the computer vision community in the last years.

Video Collection. The video sequences contained in TREK-150 have been sampled from the EPIC-Kitchens-55 (EK-55) dataset [19]. This has been done because EK-55 is currently the largest dataset for understanding human-object interactions in FPV (it provides up to 55 hours of human-object interaction examples). Thanks to its dimension, it is the only database that provides a significant amount of diverse interaction situations between various people and several different types of objects. Hence, it allowed us to select suitable diverse tracking sequences that reflect the common scenarios tackled in FPV tasks.

Bounding Box Annotations. To represent the spatial localization of objects, we employed axis-aligned bounding-boxes. This design choice for the TREK-150 benchmark is supported by the fact that this representation is largely used in many FPV pipelines [32, 34, 33, 19, 45, 83, 79]. Therefore, computing performance results based on such allows us to correlate them to the results of other FPV tasks that employ the same object representation. Hence, we can better highlight the impact that trackers would have in such contexts. Moreover, we would like to highlight the difficulty that the FPV setting poses on the development of more sophisticated annotations for object categories that appear commonly in FPV scenarios. Figure 6 shows some examples of these. The first two images from the left show the objects "cheese" and "onion" (these are considered as single objects according to the EK-55 annotations [19]) which prevent the determination of the angle for an oriented bounding-box, or an even accurate segmentation mask due to their spatial sparsity. The two images on the right present objects for which providing a segmentation is very ambiguous. Indeed, most of the pixels in the image area of the knife (third image) belong actually to foam, while the heavy motion blur happening on the object of the fourth image (where the target is a bottle) prevents the definition of the actual pixels belonging to the object. In all these scenarios, axis-aligned bounding-boxes result in robust target representations that provide a consistent delineation of the object. For these motivations, and to make representations and annotations consistent across the whole dataset, we employed such annotation representations.

Moreover, the latest progress of visual tracking algorithms on various benchmarks that use this state representation [91, 66, 35, 59, 67, 31, 41] demonstrates that it provides sufficient information about the target for consistent and reliable performance evaluation. Furthermore, using more sophisticated target representation would have restricted our analysis [86, 60, 28, 10] since the majority of state-of-the-art trackers output just axis-aligned bounding boxes [11, 23, 40, 68, 6, 7, 39, 21, 46, 80, 54, 84, 71, 53, 97, 22, 8, 93, 24, 42, 92, 18, 16, 98, 9].

Finally, we point out that the proposed axis-aligned bounding-boxes have been carefully and tightly drawn around the visible parts of the objects. Figure 7 shows some examples of the quality of the bounding box annotations of TREK-150 in contrast to the ones available in the popular OTB-100 tracking benchmark.

Frame Rate. The videos contained in TREK-150 have a frame rate of 60 FPS. This is inherited from the EK-55 dataset [19], from which videos are sampled. According to the authors [19], EK-55 has been acquired with such a setting because of the proximity of the camera point of view and the main scene (i.e. manipulated objects), which causes very fast motion and heavy motion blur when the camera wearer moves (especially when he/she moves the head).

We empirically evaluated the fast motion issue by assessing the average normalized motion happening on the frames that include fast motion (FM) (we computed them by considering the automatic procedure defined in [91, 67] to assign the FM attribute). Such a motion quantity has been computed as the distance between the center of two consecutive ground-truth bounding boxes normalized by the frame size. Considering TREK-150 with the videos at 30 FPS, such a value achieves 0.075. This is higher than the 0.068 obtained for OTB-100, the 0.033 of UAV123, or the 0.049 of NfS considered at 30 FPS. These comparisons demonstrate that the FPV scenario effectively includes challenging scenarios due to the faster motion of targets/scene. Considering the 60 FPS frame rate, the fast motion quantity of TREK-150 is reduced to 0.062, which is comparable to the values obtained in other third-person tracking benchmarks.

Sequence Labels. To study the performance of trackers under different aspects, the sequences of TREK-150 have been associated with one or more of 17 attributes that indicate the visual variability of the target in the sequence (see

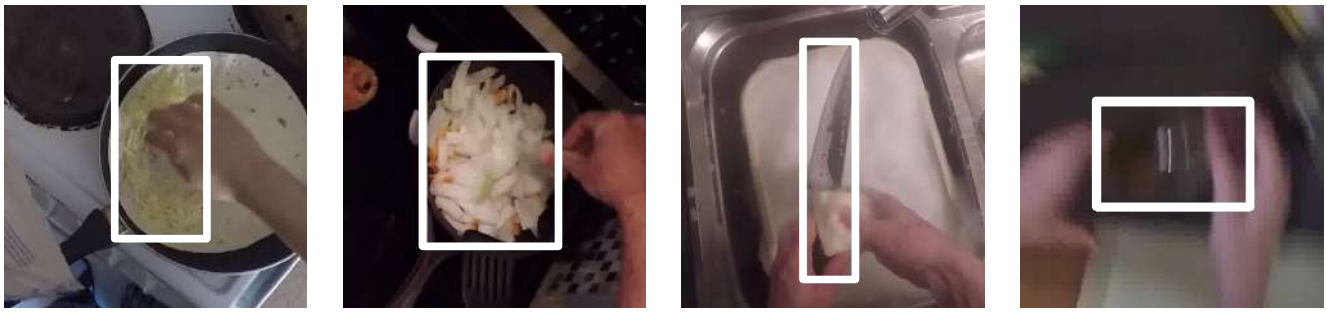

Figure 6: Examples of target objects contained in TREK-150 that are difficult to represent with more sophisticated representations (e.g. rotated bounding box or segmentation mask). The first two images from the left show objects such as "cheese" and "onion" which prevent the determination of the angle for an oriented bounding box, or an accurate segmentation mask. The last two images present objects which prevent a consistent definition of a segmentation.

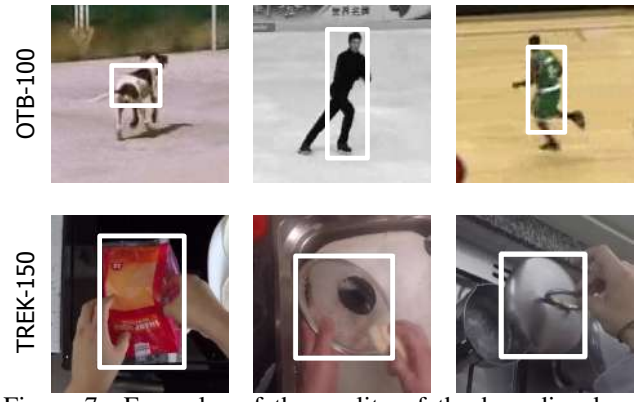

Figure 7: Examples of the quality of the bounding box annotations contained in TREK-150 in comparison with the ones available in the popular OTB-100 benchmark. TREK-150 provides careful and high-quality annotations that tightly enclose all the target objects.

Table 2 of the main paper for the details). The extended usage of this practice [91, 66, 35, 52, 67, 31, 41] showed how this kind of labeling is sufficient to estimate the trackers' performance on particular scenarios. We therefore follow such an approach to associate labels on TREK-150's videos. However, we argue that, by using this labeling setting, attention must be paid to how trackers are evaluated. The standard OPE protocol, which has been generally used to perform such evaluations, could lead to less accurate estimates. For example, it could happen that a tracker would fail for some event described by an attribute (e.g. FOC) in the first frames of a video, but that the sequence also contains some other event (e.g. MB) in the end. With the score averaging procedure defined by the OPE protocol, the low results achieved due to the first event would set low scores also for the second event, while the tracker failed just for the first one. Therefore, the performance estimate for the second attribute would not be realistic. We believe a reasonable option is to use a more robust evaluation protocol such as the multi-start evaluation (MSE). Thanks to its points of initialization which generate multiple diverse subsequences, this protocol allows a tracker to better cover all the possible situations happening along the videos, both forward and backward in time. All the results achieved on the sub-sequences are then averaged to obtain the overall scores on a sequence. We think the scores computed in this way to be more robust and accurate estimates of the real performance of the trackers. Hence, in this work, we follow such an approach to evaluate trackers over sequence attributes.

Single Object Tracking. In this paper, we restricted our analysis to the tracking of a single object per video. This has been done because in the FPV scenario a person interacts through hands with one or two objects at a time in general [19] (if a person interacts with two objects they can be still tracked by two single-object trackers). Moreover, focusing on a single object allows us to analyze better all the challenging and relevant factors that characterize the tracking problem in FPV. We believe that future work could investigate the employment of multiple object tracking (MOT) solutions [25] for a general understanding of the position and movement of all objects visible in the scene. We think that the study presented in this paper will give useful insights even for the development of such methods.

Differences With Other Tracking Benchmarks. We believe that the proposed TREK-150 benchmark dataset offers *complementary* features with respect to existing visual tracking benchmarks.

Table 1 and Figure 2(a) and (b) of the main paper show that TREK-150 provides complementary characteristics to what is available today to study the performance of visual trackers. Particularly, our proposed dataset offers different distributions of the common challenging factors encountered in other datasets. For example, TREK-150 includes a larger number of examples with occlusions (POC), fast motion (FM), scale change (SC), aspect ratio change (ARC),

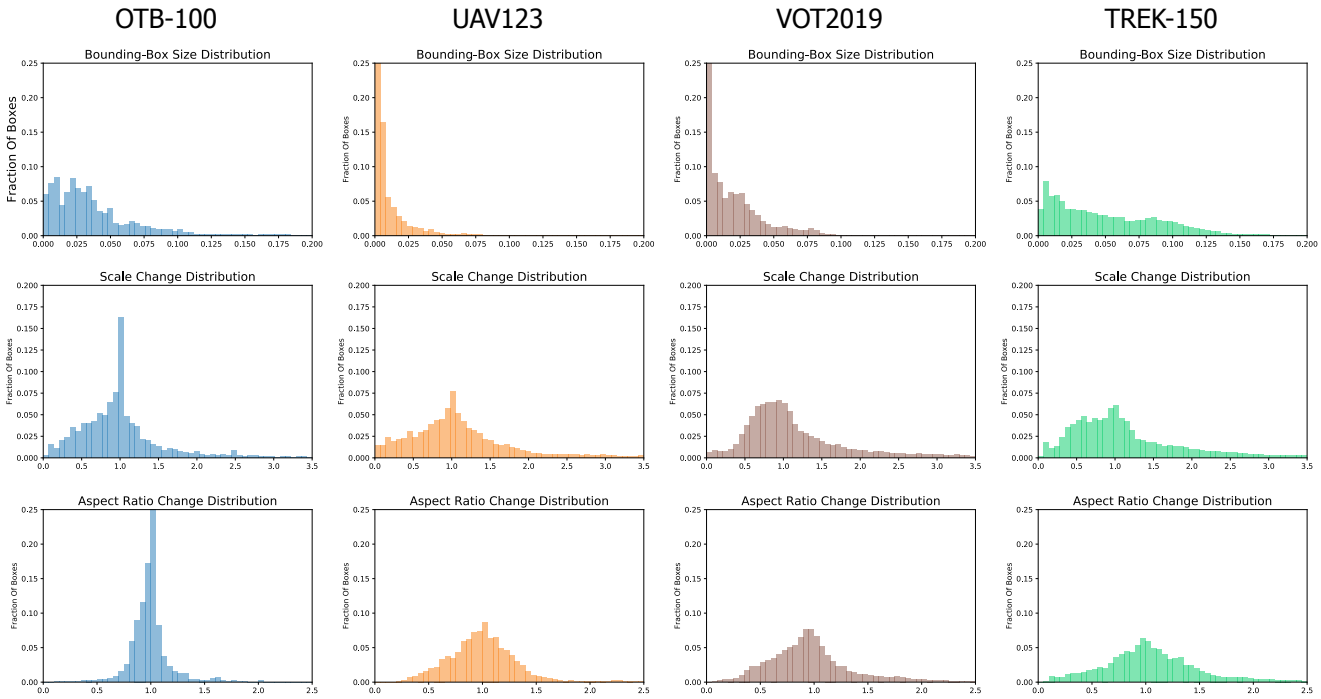

Figure 8: Comparison between TREK-150 (last column of plots) and other popular visual tracking benchmarks on the distributions computed for different bounding box characteristics. Each column of plots reports the distribution of bounding box sizes, scale changes, and aspect ratio change (the x-axis of each plot reports the range of the bounding box statistic).

illumination variation (IV), and motion blur (MB), while it provides a competitive number of scenarios for low resolution (LR), full occlusion (FOC), deformable objects (DEF), and presence of similar objects (SOB). Additionally, even though the 4 new attributes high resolution (HR), head motion (HM), one-hand interaction (1H), two-hands interaction (2H), define particular FPV scenarios, we think that they can be of interest even for the visual tracking community. For example, as shown by the second row of images of Figure 7, 1H and 2H can be considered as attributes that define different levels of occlusion, as objects manipulated with two hands generally cause more extended hiding of the targets. Besides these sequence-level features, TREK-150 offers up to 34 target categories which, to the best of our knowledge, have never been studied. As shown by the Figures 6 and 7, these objects have challenging appearances (e.g. transparent or reflective objects like lids, bottles, or food boxes) and shapes (e.g. knives, spoons, cut food) that change dramatically due to the interaction or motion induced by the camera viewer.

We additionally computed some statistics on the bounding box ground-truth trajectories contained in the proposed dataset. Figure 8 reports these distributions. For comparison, we report the distributions computed on the popular tracking benchmarks VOT2019, UAV123, OTB-100. As can be noted, our dataset exhibits different distributions, and thus offers different behaviors of the target appearances and motions. Particularly, observing the first plot of the last column, it can be noted that TREK-150 has a wider distribution of bounding box sizes, hence making it suitable for the evaluation of trackers with targets of many different sizes. Particularly, TREK-150 has a larger number of bounding boxes with greater dimension. The plot just below the first shows that TREK-150 provides more references to assess the trackers' capabilities in tracking objects that become smaller. Finally, the last plot shows a wider distribution for the aspect ratio change, showing that TREK-150 offers a large variety of examples to evaluate the capabilities of trackers in predicting the shape change of targets.

Additionally to these characteristics, we think TREK-150 is interesting because it allows the study of visual object tracking in unconstrained scenarios of *every-day* situations.

B. Tracker Details

Generic Object Trackers Details. Table 5 reports some additional information about the 31 considered generic-object trackers such as: venue and year of publication; type of image representation used; type of matching strategy; employment of target model updates; and category of tracker according to the classification of [61]. For each

Table 5: Details of the trackers involved in our evaluation. In the Image Representation column the acronyms stand for: CNN - Convolutional Neural Network; HOG - Histogram of Oriented Gradients; Pixel - Pixel Intensity; Color - Color Names or Intensity. Regarding the Matching strategy column the acronyms stand for: CF - Correlation Filter; CC - Cross Correlation; T-by-D - Tracking by Detection; Reg - Regression; Had - Hadamard Correlation. The ✓ symbol in the Model Update column expresses the target model update during the tracking procedure. The last column reports the tracking method class according to [61] (ST - Short-Term trackers, LT - Long-Term trackers).

Tracker	Venue	Image Representation	Matching	Model Update	[61] Class
MOSSE [11]	CVPR 2010	Pixel	CF	✓	ST_0
DSST [23]	BMVC 2014	HOG+Pixel	CF	✓	ST_0
KCF [40]	TPAMI 2015	HOG	CF	✓	ST_0
MDNet [68]	CVPR 2016	CNN	T-by-D	✓	ST_1
Staple [6]	CVPR 2016	HOG+Color	CF	✓	ST_0
SiamFC [7]	ECCVW 2016	CNN	CC	✗	ST_0
GOTURN [39]	ECCV 2016	CNN	Reg	✗	ST_0
ECO [21]	CVPR 2017	CNN	CF	✓	ST_0
BACF [46]	ICCV 2017	HOG	CF	✓	ST_0
DCFNet [85]	ArXiv 2017	CNN	CF	✓	ST_0
VITAL [80]	CVPR 2018	CNN	T-by-D	✓	ST_1
STRCF [54]	CVPR 2018	HOG	CF	✓	ST_0
MCCTH [84]	CVPR 2018	HOG +Color	CF	✓	ST_0
DSLT [58]	ECCV 2018	CNN	CC	✓	ST_0
MetaCrest [71]	ECCV 2018	CNN	CF	✓	ST_1
SiamRPN++ [53]	CVPR 2019	CNN	CC	✗	ST_0
SiamMask [86]	CVPR 2019	CNN	CC	✗	ST_0
SiamDW [97]	CVPR 2019	CNN	CC	✗	ST_0
ATOM [22]	CVPR 2019	CNN	CF	✓	ST_1
DiMP [8]	ICCV 2019	CNN	CF	✓	ST_1
SPLT [93]	ICCV 2019	CNN	CF	✓	LT_1
UpdateNet [96]	ICCV 2019	CNN	CC	✓	ST_0
SiamFC++ [92]	AAAI 2020	CNN	CC	✗	ST_0
GlobalTrack [42]	AAAI 2020	CNN	Had	✗	LT_0
PrDiMP [24]	CVPR 2020	CNN	CF	✓	ST_1
SiamBAN [16]	CVPR 2020	CNN	CC	✗	ST_0
D3S [60]	CVPR 2020	CNN	CF	✗	ST_0
LTMU [18]	CVPR 2020	CNN	CF/CC	✓	LT_1
Ocean [98]	ECCV 2020	CNN	CC	✗	ST_0
KYS [9]	ECCV 2020	CNN	CF	✓	ST_1
TRASFUST [29]	ACCV 2020	CNN	Reg	✗	ST_1

tracker, we used the code publicly available and adopted default parameters for evaluation purposes.

FPV Trackers Details. In this section, we provide details on the LTMU-F and LTMU-H FPV trackers considered as baselines in our study. For a better understanding, we briefly recap the processing procedure of the LTMU tracker [18]. After being initialized with the target in the first frame of a sequence, at every other frame LTMU first executes the short-term tracker DiMP [8] that tracks the target in a local area (based on the target's last known position) of the frame. The image patch extracted from the bounding box prediction of DiMP is evaluated by an online-learned verifying module which outputs a probability estimate for the target being contained in the patch. Such an estimate is employed to decide if the short-term tracker is tracking the target or not. If it is, the box predicted by the short-term tracker is given as output for the current frame. In the other case, a re-detection module is executed to search for the target in the global frame. The detector returns some candidate locations to contain the target and each of these is checked by the verification module. The candidate patch with the highest confidence is given as output and used as a new target location to reset the short-term tracker.

In our setting, we employ FPV-based detectors to implement such a re-detection module. For LTMU-F, we employed the EK-55 trained Faster R-CNN [19]. Among the many detections given as output, this module has been set to retain the first 10, considering a ranking based on the scores attributed by the detector to each detection. If no detection is given for a frame, the last available position of the target is considered as candidate location. For LTMU-H, we employ the object localization contained in the hand-object interaction detections given by the FPV version of Hands-in-contact [79] to obtain the target candidate locations. Such a solution [79] is implemented as an improved Faster R-

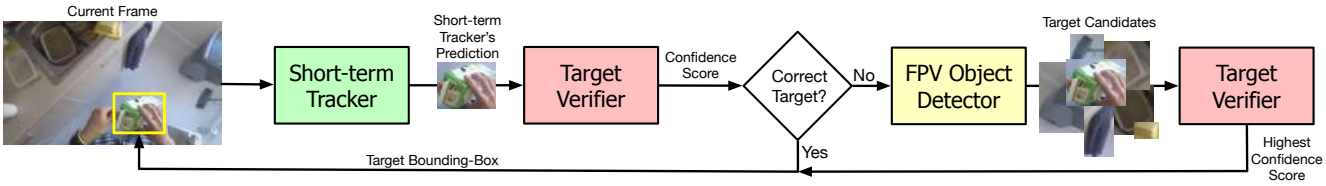

Figure 9: Visual representation of the LTMU [18] scheme performed at every frame that has been adapted for the development of the baseline FPV trackers LTMU-F and LTMU-H.

CNN which is set to learn to provide, at the same time, the localization of hands and objects, and their state of interaction. As before, if no detection is given for a frame, the last available position of the target is considered as candidate location. For both methods, the original pre-trained models (made available by the authors) which consider FPV data have been used. The described setups, which the common scheme is presented in Figure 9, give birth to two trackers that implement conceptually different strategies for FPV-based object localization. Indeed, the first solution reasons just to find objects in the scene, while the second reasons in terms of the interaction happening between the camera wearer (i.e. hands) and the objects. We would like to remark that other FPV trackers (such as the ones described in Section 2 of the main paper) have not been tested on TREK-150 because their implementations are not available.

Implementation Details. The evaluations were performed on a machine with an Intel Xeon E5-2690 v4 @ 2.60GHz CPU, 320 GB of RAM, and an NVIDIA TITAN V GPU. We considered the Python publicly available implementations of each tracker and adopted default parameters. Annotations, results of the trackers, and code are available at https://machinelearning.uniud.it/datasets/trek150/.

C. Experimental Details

Details On The Generalized Robustness. The robustness measure has been first introduced in [51]. This metric was defined as the number of drifts (i.e. the complete non-overlap between predictions and ground-truths) performed by a visual tracking algorithm. In the last iteration of the VOT challenge [47], such a robustness measure has been revised and defined as the extent of a tracking sequence before the tracker's failure. Such an extent is determined as the number of frames positively tracked normalized by the total number of frames in the sequence. The failure event is triggered when the overlap between the predicted and ground-truth bounding-boxes becomes lower than a fixed threshold (the value 0.1 is used in [47]). In simpler words, this measure expresses the fraction of a tracking sequence that is correctly tracked from its beginning. We think this measure is of special interest to the FPV community. Indeed, it can assess the ability of a tracker to maintain reference in time to the target objects. Since many FPV tasks are devoted to understand the action performed by the camera viewer or its interaction with objects [32, 34, 33, 19, 73], having solutions capable of maintaining temporally longer references to target nouns can be advantageous to model such events. However, we believe that having a single fixed threshold is restrictive, as different applications can make different assumptions on the concept of tracking failure. Therefore, following [91] which proposed to evaluate trackers with plots computed after thresholding bounding-box overlaps with different values, we propose to build a plot considering different overlap thresholds for the determination of failure in the robustness measure [47]. This leads to the creation of the Generalized Success Robustness Plot (Figure 3(c) of the main paper) which reports the different robustness scores for the different thresholds. The latters have been studied just in the range [0, 0.5] because it is common practice, in the computer vision literature, to consider overlaps greater than 50% as positive predictions. Notice that failures could be also defined in terms of center error. In this paper, we focused on overlap-based failures since bounding-box overlap has been shown to be superior for target localization accuracy [99], but future work will investigate the employment of the center error as this kind of bounding box distance is used in FPV tasks [79]. Moreover, to compare trackers with a single score, following [91] and [67], we compute the AUC of the Generalized Success Robustness Plot which we refer as to generalized success robustness (GSR). This value expresses the average of all the scores obtained with the different thresholds. In other words, the GSR score expresses the average successful extent of the predictions of a tracker.

Details On The Evaluation Protocols. In this section, we give further details on the experimental protocols used for the execution of the trackers.

The one-pass evaluation (OPE) protocol, which is detailed in [91], consists of two main stages: (i) initializing a tracker with the ground-truth bounding box of the target in the first frame; (ii) let the tracker run on every subsequent frame until the end of the sequence and record predictions to be considered for the evaluation. For each sequence, predictions and ground-truth bounding boxes are compared ac-

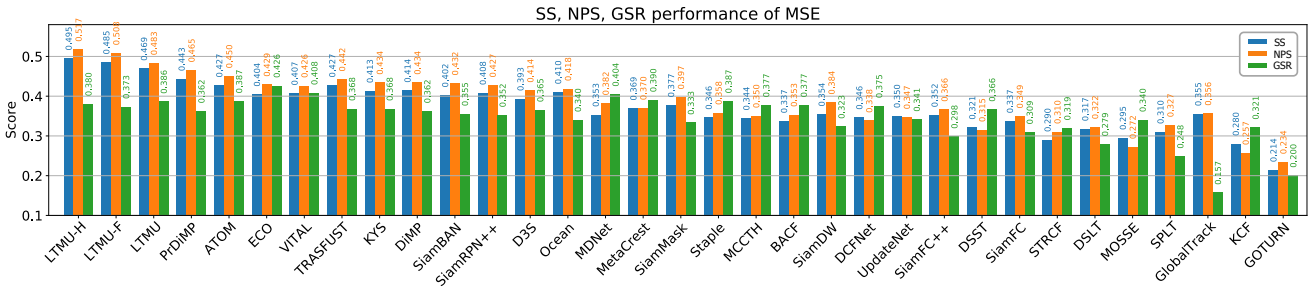

Figure 10: SS, NPS, and GSR performance of the 33 benchmarked trackers on the proposed TREK-150 benchmark under the MSE protocol. The general low performances confirms the conclusions achieved with the OPE protocol.

cording to the employed measures (only for frames where ground-truths are present) to obtain the performance scores. The overall scores, presented in brackets in Figure 3 of the main paper, are obtained by averaging the scores achieved for every sequence.

For the implementation of the multi-start evaluation (MSE) protocol, we followed the details given in [47]. For each sequence, different points of initialization (called anchors) separated by 2 seconds (in our setting every 120 frames) are defined. Anchors are always set in the first and last frame of a sequence. Some anchors are shifted forward for a few frames to obtain a more consistent bounding-box for tracker initialization. A tracker is run on each of the sub-sequences yielded by the anchor (in total 1032 sub-sequences are generated), either forward or backward in time depending on the longest sub-sequence the anchor generates. The tracker is initialized with the ground-truth in the first frame of the sub-sequence and let run until its end. Then, similarly as for the OPE, predicted and ground-truth bounding boxes are compared to obtain the performance scores for each sub-sequence. Scores for a single sequence are computed by a weighted average where the scores of each sub-sequence are weighted by its length (as number of frames). Similarly, the overall scores for the whole dataset (which are shown in Figure 10) are obtained by a weighted average where each sequence's score is weighted by the number of frames in that sequence.

The real-time evaluation (RTE) protocol has been implemented following the details in [48, 55]. Similar to the OPE protocol, a tracker is initialized with the ground-truth in the first frame of a sequence. Then the tracker is presented with a new frame only after its execution over the previous frame has finished. The new presented frame is the last frame available for the time instant in which the tracker becomes ready to be executed, considering that frames occur regularly according to the frame rate of the video. In other words, all the frames occurring in the time interval between the start and end time instants of the tracker's execution are skipped. For all the frames skipped, the last bounding box given by the tracker is used as location for the target in such frames. The sequence and overall scores are ultimately obtained as for the OPE protocol.

Experiments On The Impact of Trackers In FPV. In this section, we report more details on the experiments performed to evaluate the impact of trackers in FPV tasks (presented in the paragraph "Do Trackers Already Offer Any Advantage in FPV?" of Section 6 of the main paper.)

In the first experiment, we assessed the capabilities of continuous object localization of an object detector, as this method is usually exploited in many FPV pipelines. We executed the EK-55 trained Faster-R-CNN [19] on all the frames of TREK-150, by recording in each frame the bounding box of the detection having the same class of the target and the highest confidence score. Such bounding box predictions were then compared to the ground truth annotations using the considered tracking evaluation measures. This experimental strategy respects the evaluation procedure of the OPE protocol. In this way, we can compare the performance of the tracking approach and the detection approach in providing localization and temporal reference of/to objects.

In the second experiment, we evaluated the impact of trackers in a video-based hand-object interaction detection setting. Since this paper is focused on objects (visual object tracking), we restricted our study in evaluating the detection of the objects involved in the interactions. To this aim, we first built tracks of hand-object interactions over the sequences of TREK-150. The hand-object interaction detector Hands-in-contact [79] has been executed to obtain sparse interaction detections that involved the object defined by the ground-truths of TREK-150. Clusters of detections have been then set to form separate tracks if the interval between two detections was longer than 30 frames. The missing detections within a cluster have been filled with the TREK-150's object ground-truth bounding boxes and the most frequent interaction state (i.e. if the object was in interaction with the left hand, the right hand, or both) appearing in the cluster. Once had these references, we ran the trackers in an OPE-like fashion. Each tracker was initialized in the first

Table 6: Performance achieved by the 33 benchmarked trackers on TREK-150 using the RTE protocol.

Tracker	FPS	SS	NPS	GSR
Ocean	21	0.365	0.358	0.294
SiamBAN	24	0.360	0.366	0.313
SiamRPN++	23	0.362	0.356	0.293
PrDiMP	13	0.352	0.349	0.243
DiMP	16	0.336	0.331	0.224
SiamMask	23	0.335	0.333	0.298
SiamFC++	45	0.330	0.331	0.308
SiamDW	32	0.327	0.334	0.317
KYS	12	0.327	0.317	0.237
ATOM	15	0.319	0.312	0.179
UpdateNet	21	0.311	0.297	0.295
DCFNet	49	0.299	0.286	0.335
TRASFUST	13	0.296	0.270	0.185
SiamFC	34	0.293	0.295	0.280
LTMU	8	0.284	0.257	0.169
D3S	16	0.276	0.263	0.182
BACF	9	0.276	0.262	0.234
SPLT	8	0.265	0.247	0.203
STRCF	10	0.264	0.250	0.218
DSLT	7	0.260	0.234	0.211
ECO	15	0.252	0.231	0.173
GlobalTrack	8	0.253	0.227	0.139
MCCTH	8	0.251	0.231	0.232
Staple	13	0.249	0.236	0.169
GOTURN	44	0.247	0.242	0.119
MOSSE	26	0.227	0.190	0.141
LTMU-H	4	0.213	0.174	0.161
MetaCrest	8	0.207	0.175	0.165
LTMU-F	4	0.205	0.161	0.162
VITAL	4	0.204	0.165	0.158
DSST	2	0.191	0.145	0.161
KCF	6	0.186	0.157	0.177
MDNet	1	0.185	0.140	0.161

frame of a track with the object detection given by Hands-in-contact [79], and then let run for the other frames of the track. We then evaluated the performance in a track by the normalized count of frames having intersection-over-union ≥ 0.5 with the object's ground-truth. The overall result is obtained by averaging the outcomes of all tracks. This experimental procedure gives us an estimate of the accuracy of the hand-object interaction detection system if trackers would have been included in its pipeline. More interestingly, it allows also to build a ranking of the trackers based on the results of a downstream application.

D. Additional Results

MSE Protocol Results. Figure 10 reports the overall performance of the 33 benchmarked trackers on TREK-150 using the MSE protocol. The overall low performances of all the trackers confirm the conclusions achieved using the OPE protocol. The FPV setting introduces challenging factors for current visual trackers.

Qualitative Examples The first 7 rows of images of Figure 11 present qualitative results of 10 of the generic-object trackers in comparison with the ground-truth (which is identified by the white rectangles). The action performed by the camera wearer is also reported for each sequence. The remaining 4 rows show the qualitative performance of the FPV baseline trackers LTMU-F and LTMU-H in comparison with LTMU and the ground-truth. For a better visualization, a video can be found at https://youtu.be/oX1nICHgEJM.

Table 7: Performance of the offline trackers SiamFC and SiamRPN++ on a subset of 50 sequences of TREK-150 without and with fine-tuning on the remaining 100 videos.

Tracker	Fine-tuning	OPE			MSE		
		SS	NPS	GSR	SS	NPS	GSR
SiamFC	✗	0.311	0.332	0.317	0.307	0.317	0.307
	✓	0.267	0.275	0.278	0.287	0.305	0.292
SiamRPN++	✗	0.384	0.395	0.377	0.367	0.385	0.333
	✓	0.348	0.407	0.313	0.336	0.406	0.314

Per Attribute/Action Results. Figure 12 presents the SS, NPS, and GSR scores achieved by the 33 trackers considering the attributes assigned to sequences. Similarly, Figures 13 and 14 report the results for the whole batch of trackers with respect to action verbs and target nouns.

RTE Protocol Results. Table 6 reports the FPS, SS, NPS, and GSR performance of all 33 benchmarked trackers obtained using the RTE protocol. As stated in the main paper, offline siamese trackers emerge as the best solution in this scenario. Online deep discriminative trackers achieve comparable results in SS and NPS, but demonstrate a larger drop in performance in the GSR score, showing that online learning mechanisms influence this performance in the real-time setting.

Adaptation Of Offline Trackers. Many current visual trackers employ deep learning architectures. Among these, trackers based on siamese neural networks emerged as the most popular approaches nowadays. These trackers are said to be offline (e.g. SiamFC [7], SiamRPN++ [53], SiamMask [86], SiamBAN [16]) because they are trained to track objects on large-scale tracking dataset [26, 67, 41, 31], and do not use online adaptation mechanisms at test time. In our evaluation, such trackers have been employed as they are described and trained in their original paper. Given their generally low performance, one could wonder how these trackers perform if knowledge about the FPV domain is exploited for learning. Our TREK-150 dataset, which is designed to evaluate the progress of visual tracking solutions in FPV, does not provide a large-scale database of learning examples as needed by these methods. Instead, it well aligns with real-world datasets where millions of frames are not available for training. In such scenarios, the reasonable options the machine learning community suggest are

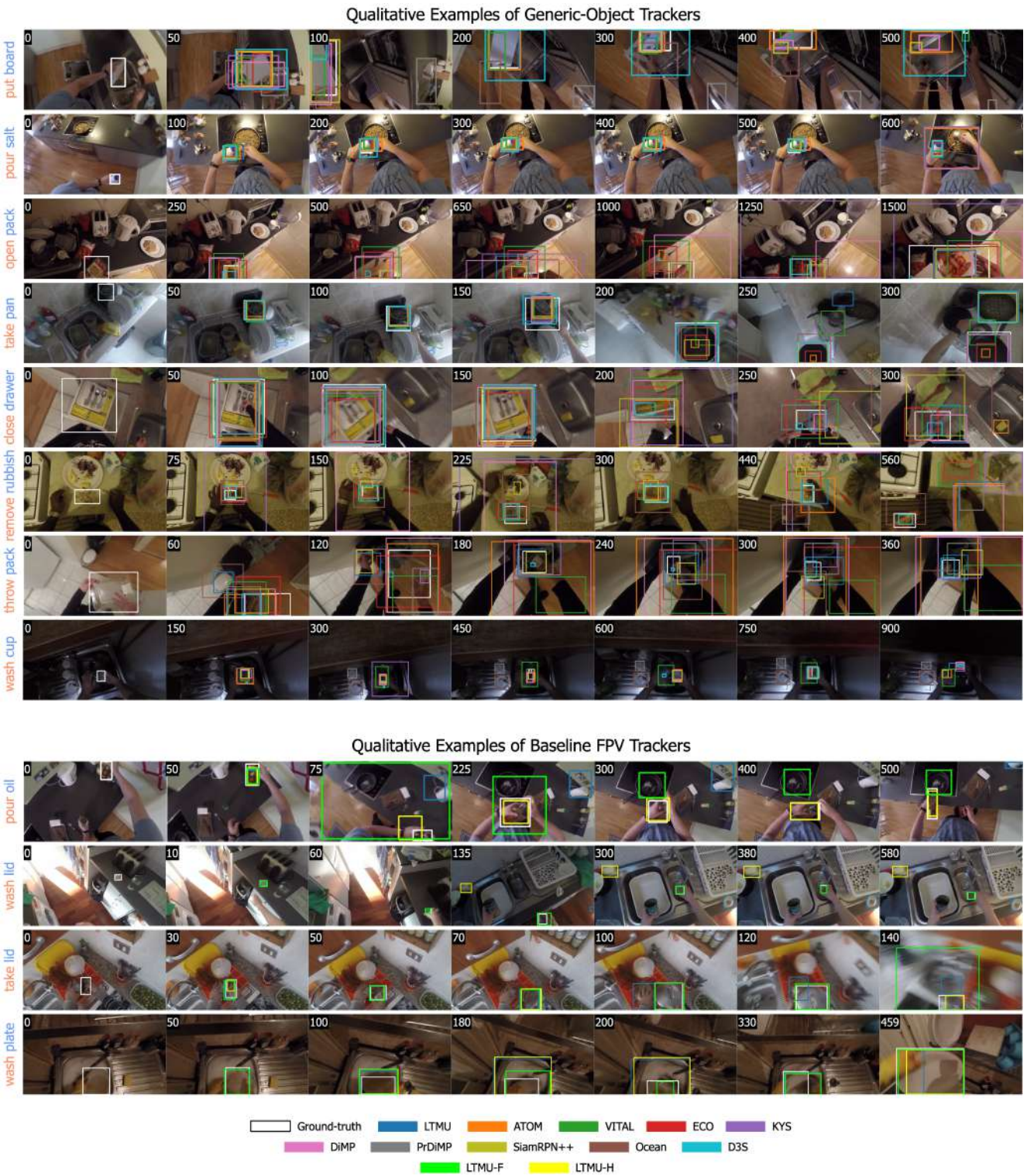

Figure 11: Qualitative results of some of the studied trackers on the proposed TREK-150 dataset. The first 7 rows of images show the qualitative performance of 10 of the selected generic-object trackers, while the last 4 rows show the results of the baseline FPV trackers LTMU-F and LTMU-H in comparison with LTMU. For a better visualization, a video can be found at https://youtu.be/oX1nICHgEJM.

to use trackers as they are because of their general knowledge, or to adapt them through fine-tuning using a smaller training set. We tried the second strategy by randomly splitting TREK-150 in a training and test set of 100 and 50 videos respectively. We fine-tuned the popular offline trackers SiamFC and SiamRPN++ on the training set according to their original learning strategy. We then tested the fine-tuned versions on the test set and the results are reported in Table 7. It shows that fine-tuning leads to substantial overfitting that cause the performance to drop in general. These outcomes prove the decision to evaluate offline trackers as they are is the right one given the current lack of large-scale FPV tracking datasets. Moreover, given the overall results presented in this paper, we hypothesize that visual tracking in FPV will require more than just large-scale training. We hope the results presented in this section will encourage the community to work on domain adaptation techniques for offline trackers that are currently starting to be investigated [30].

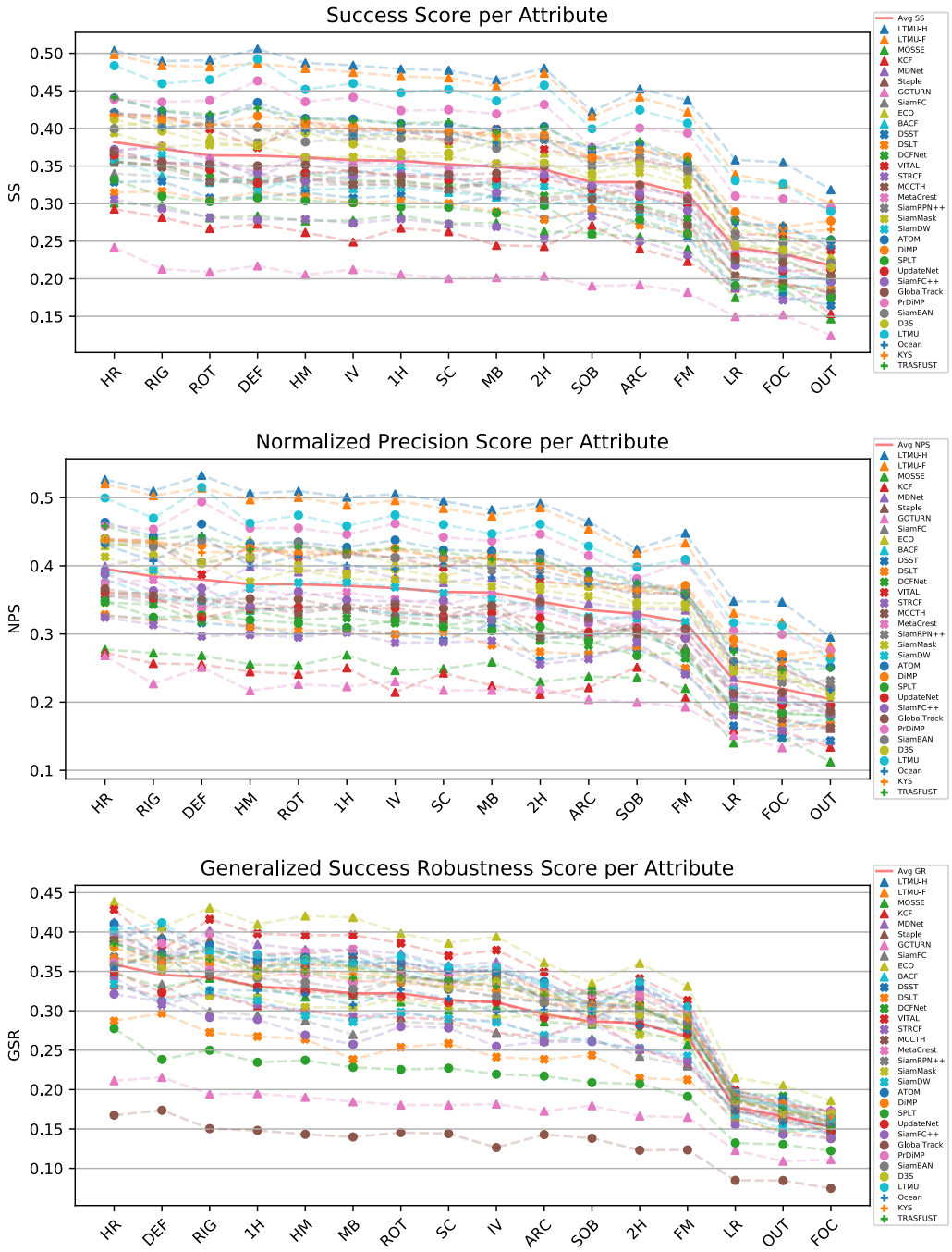

Figure 12: SS, NPS, and GSR results per sequence attribute achieved by the 33 benchmarks on the TREK-150 benchmark.

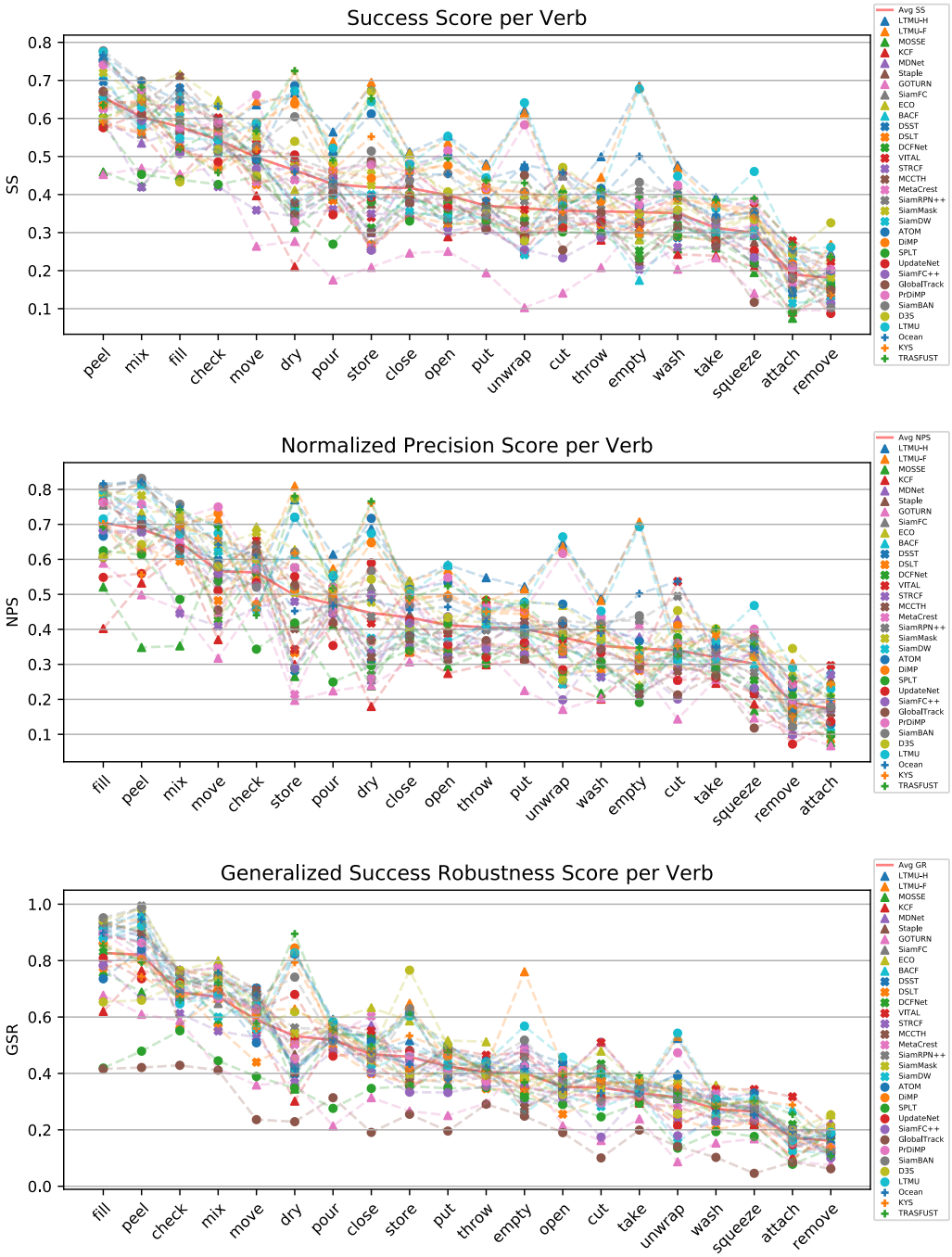

Figure 13: SS, NPS, and GSR results achieved by the 33 benchmarks on the TREK-150 benchmark considering each verb associated to the action performed by the camera wearer.

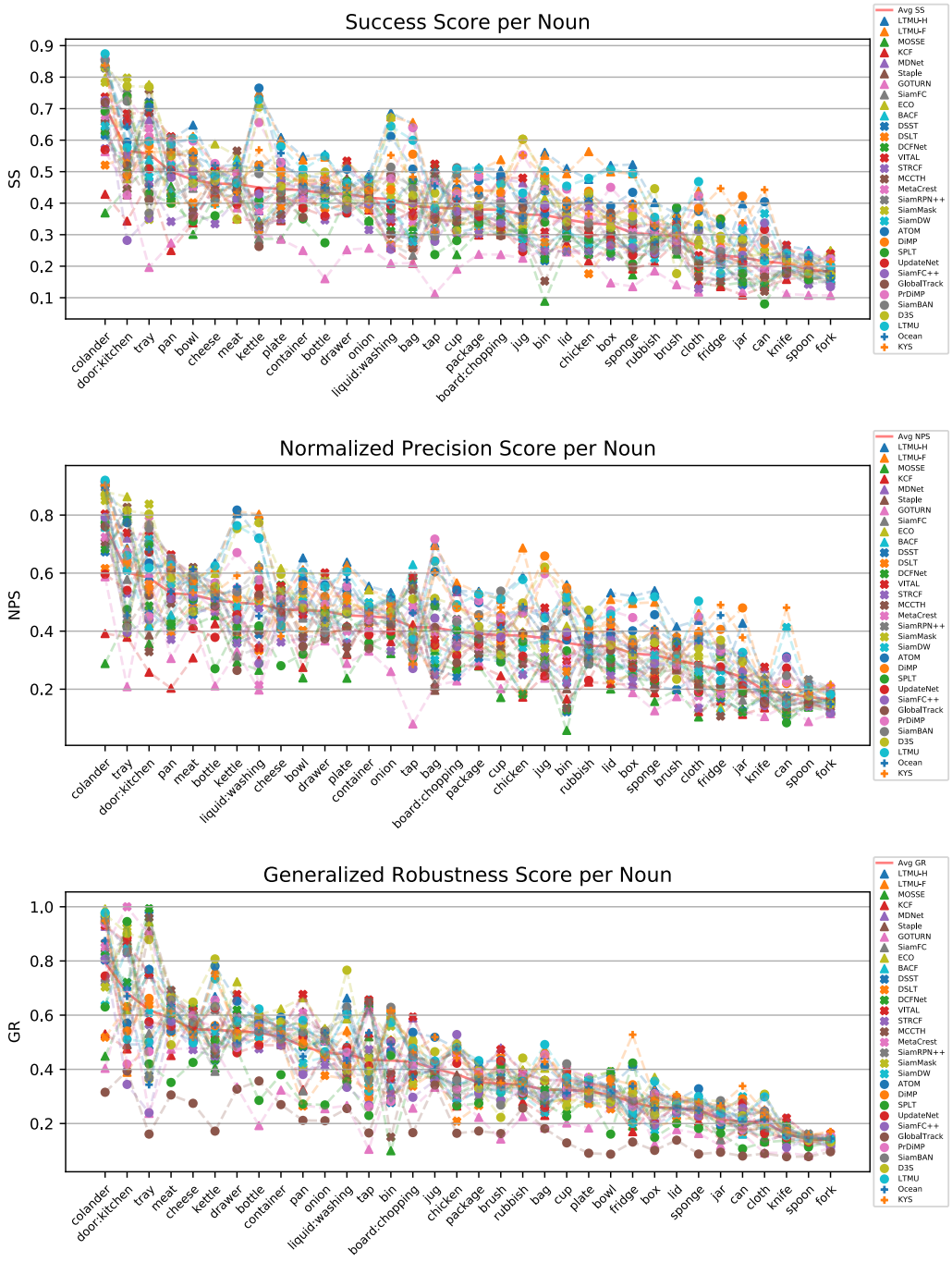

Figure 14: SS, NPS, and GSR results achieved by the 33 benchmarks on the TREK-150 benchmark considering the different target categories.