Tracking Skiers from the Top to the Bottom

Supplementary Document

Matteo Dunnhofer, Luca Sordi, Niki Martinel, Christian Micheloni
Corresponding author e-mail: matteo.dunnhofer@uniud.it

A. Further Details on SkiTB

This section provides additional information about the data contained in SkiTB as well as further motivations behind its construction.

To avoid confusion, we state that, in the scope of this paper, a skiing course is considered as a path or track down a mountain slope that an athlete should follow to complete his/her performance. It should not be confused with a course taken to learn how to ski.

A.1. Bounding-box Representation

As stated in the main paper, the motivation behind the employment of bounding-boxes is grounded on the fact that such a representation is sufficiently informative for the computational processes performed by higher-level skiing performance understanding tasks [1, 52, 53, 78]. The aforementioned pipelines simply require a rectangle highlighting the area covered by the skier's appearance. Compared to the more complex segmentation masks [47, 48], the four-value representation of bounding-boxes demands less computational resources, thus enabling the development of more efficient methods. Additionally, the choice of including the appearance of the skiing equipment within the labeled bounding-box is guided by the common working mechanism of the aforementioned solutions, which necessitate a bounding-box encompassing both the athlete's body and equipment.

A.2. Details on the Visual Attribute Labels

Table 7 presents the description of the attributes assigned to the SC clips. The attributes have been introduced to cluster the tracking performance depending on the visual variability events occurring on the target object. This evaluation approach of assigning per-video labels is well-established in the visual object tracking community [24, 30, 34, 42, 59, 81] and was shown to be sufficiently robust to estimate the trackers' performance in particular scenarios. Among the many attributes present in the literature, we selected 10 that well represent the variability of the skiing domain. The labels have been associated with SC clips of the date-based training-test split because the SC experimentation setting allows a tracker to cover the situations happening during the skier's descent in a more complete and consistent way [24, 47]. Figure 8 shows the distribution of the SC clips according to the labels. In SkiTB, the labels SC, ARC, FM, and LR, have been assigned by an automatic procedure as described by [30, 81]. The presence of situations identified by the other attributes has been visually assessed and annotated by our research team.

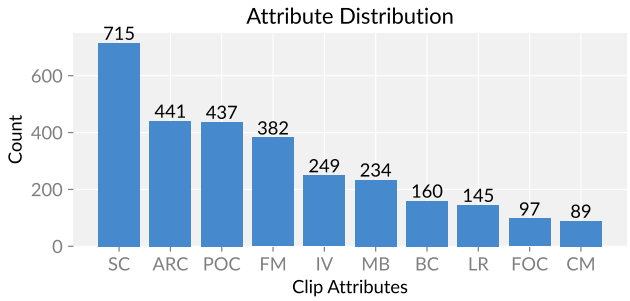

Figure 8. **Distribution of the clip attributes.** The plot shows the number of single-camera (SC) clips associated with each of the attributes introduced to characterize the visual variability of the target, as in [24, 30, 59, 81]. The application domain of skiing videos presents a large number of scale changes (SC), followed by a substantial number of partial occlusions (POC), changes in the aspect ratio (ARC), and fast motions (FM).

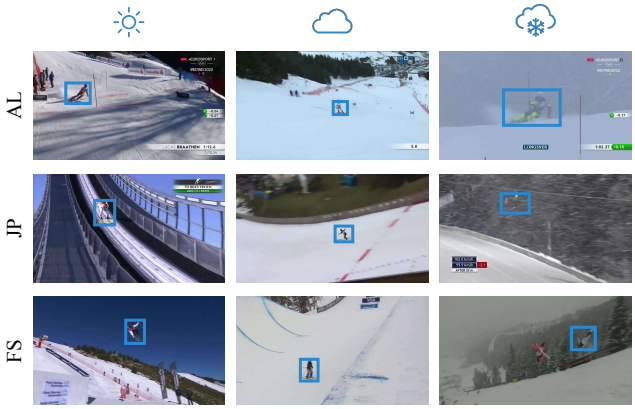

Figure 9. **Winter weather conditions.** Skiing takes place in winter environments, subjecting athletes to extreme weather conditions that introduce unique image characteristics when captured on camera. For instance, sunny conditions can create shadows, resulting in significant variations in target illumination. Cloudy weather leads to "flat light" conditions, reducing image contrast, while snowfall or rain further diminishes visibility. The SkiTB includes weather condition labels for each MC video.

Table 7. **Selected sequence attributes associated to single-camera (SC) clips.** This table gives the formal definition of the selected clip attributes according to previous research in generic visual object tracking [24,30,81]. On a side, we give an interpretation of each definition w.r.t. our application domain.

Attribute	Definition	Application-specific Interpretation
CM	Camera Motion: an abrupt camera motion can be seen in the video clip.	The camera operator moves the camera fast to keep the skier in the field of view.
SC	Scale Change: the ratio of the bounding-box area of the first and the current frame is outside the range [0.5, 2].	The size of a skier's appearance changes considerably during the video (e.g. by zooming in/out on the target).
BC	Background Clutter: the target has a similar appearance w.r.t. the surrounding background.	The appearance of the athlete's suit and equipment confounds with the elements in the background.
ARC	Aspect Ratio Change: the ratio of the bounding-box aspect ratio of the first and the current frame is outside the range [0.5, 2].	The ratio between the height and width of the athlete changes (e.g. due to complex body poses).
IV	Illumination Variation: the area of the target bounding-box is subject to light variation.	The appearance of the target skier changes due to particular lightning conditions (e.g. passing through slope areas under shadow).
POC	Partial Occlusion: the target is partially occluded in the video.	Part of the skier is hidden by another item (e.g. by a gate in AL).
MB	Motion Blur: the target region is blurred due to target or camera motion.	The appearance of the skier is blurred due to its fast motion or the fast motion of the camera.
FM	Fast Motion: the target bounding-box has a motion change larger than its size.	The skier moves fast during the descent on the course.
FOC	Full Occlusion: the target is fully occluded in the video.	The skier is completely occluded by another item in the field of view (e.g. by a kicker in FS).
LR	Low Resolution: the area of the target bounding-box is less than 1000 pixels in at least one frame.	The skier appears small due to a low level of camera zoom.

A.3. Details on the Weather Labels

The weather labels have been associated with each MC video because the weather condition generally remains the same across all the location in which the skiing competition takes place. The labeling of the conditions was performed by our team by analyzing the condition visible in the video. Such a label was also checked to match the one reported on the official result list available on the FIS database [70]. The labeling generated the following weather labels: "Clouds", "Fog", "LowClouds", "MostlyCloudy", "Overcast", "PartlyCloud", "Raining", "Snowing", "Sunny", "Clear". In order to have a larger number of samples for the experiments, such labels have been clustered into three categories: "Sunny", "Cloudy", and "Harsh" weather. Figure 9 gives examples of the image condition such weather condition cause. In total, SkiTB provides 191 videos associated with "Sunny", 66 with "Cloudy", and 43 associated with "Harsh". After the date-based training-test split, the test set used to compute the results in Figure 7 has 80 "Sunny", 26 "Cloudy", and 14 "Harsh" videos.

Details on the Training-Test Splits. Table 8 shows some statistics of the videos present in the three different training-test splits generated to train and test skier-specific trackers under different application conditions. The splits have been generated to maintain a balanced distribution across the skiing disciplines and sub-disciplines while aiming to keep condition-specific disjoint partitions and respect as close as possible a 60-40 ratio.

B. Details about the Trackers

In this section, we give some more details of the implementation of the selected trackers.

B.1. Generic Object Trackers

The generic object trackers have been selected to be representative of the state-of-the-art solutions in the years between 2010-2023. They have been implemented by exploiting the code originally provided by the authors along with pre-trained weights. The original hyper-parameter values leading to the best and most likely generalizable instances of all the trackers have been set. Those trackers that do not output a confidence score, were modified to return an always-confident score of 1.0.

B.2. Skier-specific Trackers

YOLO-SORT. The YOLO-SORT tracker implements a tracking-by-detection approach inspired by multiple object tracking [5, 22]. At each frame of a video, this baseline first detects skiers with an YOLOX instance [35][3] and then exploits the Simple Online and Realtime Tracking method (SORT) [5] to associate the new detections with previously memorized tracklets. The YOLOX instance was trained on all the frames and the associated bounding-box annotations of SkiTB's training set defined by the date-based split, by mostly default hyper-parameters. The only changes made are relative to the batch size, set to 16, and the number of training epochs, set to 25. 10% of the training videos were considered to build the set of validation images. The model instance achieving the highest Average Precision (AP) on such a subset was retained for inference during tracking. The SORT module is initialized in the first frame with the given skier's bounding-box. At every other time step, it is given in input all the detections given by YOLOX and returns a new set of tracks. As output, we retain the bounding-box associated to the track initialized in the first frame.

[3] https://github.com/ultralytics/yolov5

Table 8. **Statistics of SkiTB's training and test splits.** The following table reports some statistics of the three split that have been created to evaluate the capability of learning-based trackers in generalizing to different application conditions. For generalizing to new performances, the date associated to the videos has been used as splitting condition; for the generalization to unseen athletes, the athlete IDs; to generalize to unseen courses, the course's location information.

Generalization Condition	New performances		Unseen athletes		Unseen courses	
Split	Train	Test	Train	Test	Train	Test
# MC videos	180	120	176	124	182	118
# SC videos	1215	804	1151	868	1238	781
# frames	212793	140185	202052	150926	213093	139885
avg MC video seconds	39	39	38	41	39	40
avg SC video seconds	5.8	5.8	5.9	5.8	5.7	6.0
# sub-disciplines	11	11	11	11	11	11
# athletes	136	90	118	78	138	95
# athlete genders (M, W)	(84, 52)	(53, 37)	(70, 48)	(47, 31)	(83, 55)	(56, 39)
# athlete nationalities	21	22	22	21	23	20
# locations	124	61	122	92	99	62
# location countries	21	20	23	23	22	20

Algorithm 1 Pseudo-code of the procedure implemented by the proposed STARK$_{\texttt{SKI}}$ while running on a video.

1: *// Consider video $\mathcal{V}$ and ground-truth box b_0*
2: *// Trackers initialization*
3: Initialize STARK$_{\texttt{FT-SC}}$with F_0 and b_0
4: Initialize STARK$_{\texttt{FT}}$with F_0 and b_0
5: $t \leftarrow 1$
6: **repeat**
7: $b_t, c_t \leftarrow$ Run STARK$_{\texttt{FT-SC}}$ on F_t
8: **if** $c_t \leq 0.5$ **then**
9: $b_t, c_t \leftarrow$ Run STARK$_{\texttt{FT}}$ on F_t
10: **if** $c_t > 0.5$ **then**
11: *// STARK$_{\texttt{FT-SC}}$ re-initialization*
12: Re-initialize STARK$_{\texttt{FT-SC}}$ with F_t and b_t
13: **end if**
14: **else**
15: *// Compute bounding-box for STARK$_{\texttt{FT}}$ relocalisation*
16: $S \leftarrow \frac{H}{5.0}$ // 5.0 is STARK$_{\texttt{FT}}$ search area's factor
17: $x_t^* \leftarrow clip(x_t, \frac{H}{2}, W - \frac{H}{2})$
18: $y_t^* \leftarrow \frac{H}{2} - \frac{S}{2}$
19: $b_t^{(R)} \leftarrow [x_t^*, y_t^*, S, S]$
20: Use $b_t^{(R)}$ to reset STARK$_{\texttt{FT}}$'s box used to compute the search area location
21: **end if**
22: Return b_t, c_t as output for F_t
23: $t \leftarrow t + 1$
24: **until** $t = T$

STARK$_{\texttt{FT}}$. The STARK$_{\texttt{FT}}$ baseline implements a fine-tuned version of the generic object tracker STARK (STARK-ST50) [82]. To implement this tracker we exploited the publicly available code[4] and adapted the model's tracking ability by fine-tuning on SkiTB's training set, according to STARK's original training strategy. Mostly default hyper-parameters have been kept, except for the number of epochs in stage-one training, which has been set to 200.

STARK$_{\texttt{SKI}}$. The pseudo-code of the procedure implemented by the proposed skier-optimized tracking baseline STARK$_{\texttt{SKI}}$ is given in Algorithm 1. The procedure is composed of two skier-specific instances of STARK (STARK-ST50) [82]. The first one, which we refer to as STARK$_{\texttt{FT-SC}}$, is a modified version of STARK that, at every frame, computes the target bounding-box by exploiting a higher-resolution search area located around the previous target location. This is achieved by reducing the search area factor from the original value of 5.0 to 3.0 (we determined the value 3.0 by experiments) and fine-tuning as done for STARK$_{\texttt{FT}}$. In this way, we reduce the amount of background information present in the search area, thus increasing the resolution of the target skier's appearance and making the tracker predict more accurate bounding-boxes during single-camera tracking. Given the more limited search area, STARK$_{\texttt{FT-SC}}$ performs better just in such conditions where the target and camera motion are stable and consistent across consecutive frames. In the other cases, i.e. in those frames where STARK$_{\texttt{FT-SC}}$ is not confident in tracking the target (lines 8-13 of Algorithm 1), we exploit a STARK$_{\texttt{FT}}$ instance configured as described in the previous paragraph. This instance keeps the original search factor with a value of 5.0 and thus is able to look for the target in a larger frame area. The execution of this STARK$_{\texttt{FT}}$'s instance is generally triggered after a camera shot-cut and during the complete occlusion of the target. We empirically found beneficial to set the search area size of this instance to match the frame's height, by modifying the bounding-box values that are used to compute the search area at the next frame (lines 16-20 of Algorithm 1). The position of such a box is set to be the latest confident box position predicted by STARK$_{\texttt{FT-SC}}$, clipped to make the search area not fall outside of the frame. Whenever STARK$_{\texttt{FT}}$ finds confidently the target again, its predicted bounding-box and the respective frame are used to re-initialize STARK$_{\texttt{FT-SC}}$. We found the re-initialization to work better than just relocating STARK$_{\texttt{FT-SC}}$ on the STARK$_{\texttt{FT}}$'s predicted bounding-box.

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

C. Details on the Evaluation

In this section, we explain and motivate in more detail the evaluation procedures implemented.

Evaluation Protocols. As mentioned in the main paper, to run a tracker, we employed the OPE protocol introduced in [81] which implements the most realistic way to run a tracker in practice. The protocol consists of two main stages: (i) initializing a tracker with a bounding-box of the target in the first frame of the video; (ii) letting the tracker run on every subsequent frame until the end and recording predictions to be considered for the evaluation. To obtain performance scores for each sequence, predictions and ground-truth bounding-boxes are compared according to some distance measure. The overall scores are obtained by averaging the scores achieved for every sequence. As in the default OPE, we use the ground-truth bounding-box for initialization to evaluate the trackers in the best possible conditions, i.e. when accurate information about the target is given. However, many deployment conditions do not allow human labeling but instead require a completely automatic athlete localization system (e.g. real-time skiing performance analysis during broadcasting). To evaluate trackers in similar conditions, we use an object detector to predict the initial skier bounding box. Thus, we consider a version of the OPE protocol where each tracker is initialized in the first frame in which the YOLOX detector's [35, 44], fine-tuned for skier localization, provides a bounding-box prediction with confidence score ≥ 0.5. The fine-tuning of this detector was performed in the same way as for YOLOSORT mentioned before.

Performance Measures. To quantify the distance between the predicted and temporally-aligned ground-truth bounding-boxes, we used different measures. As general tracking accuracy indicators, we employed the metrics defined by [54] for long-term tracking problems: Precision, Recall, and F-Score. Due to the generally long video observation and presence of multiple occlusions, our problem of interest is related to such a research framework. Now we explain the meaning of such metrics in relation to our application case. The Precision (Pr $\uparrow$) measures the average amount of correctly tracked ground-truth bounding-boxes where the tracker is confident, with different thresholds used to determine the conditions of correct and confident prediction. In our case, the Pr $\uparrow$ score determines the average coverage of the skier's position on the portion of skiing performance observation on which the tracker is confident. For example, a Pr $\uparrow$ score of 0.8 tells that an algorithm correctly localizes the athlete for the 80% of the bounding-box predictions that are given with high confidence. The Recall (Re $\uparrow$) instead measures the average amount of correctly tracked ground-truth bounding-boxes, regardless of the tracker's confidence. In our context, such a score determines the average coverage of the position of the skier throughout the whole skiing performance. For instance, a Re $\uparrow$ score of 0.8 gives that the algorithm correctly localizes the athlete for 80% of the skiing performance appearing in the video. The F-Score (F-Score $\uparrow$) provides a single aggregating score that incorporates both the previous measures. The best value across the different confidence thresholds is retained.

In addition to those metrics, we exploited the Generalized Success Robustness (GSR $\uparrow$) [23, 24] which reports the fraction of continuous successful tracking before the tracker is lost, measured as the temporal index of the first wrong prediction normalized by the number of frames in the video. In the context of this application domain, such a metric reports the percentage of continuous coverage of the skier's performance before the target is lost by the tracking algorithm. The original metric [24] is strict because it considers just the first wrong prediction to determine the tracker's failure time step. Other work [47] suggested a softer version of such a measure. If the algorithm gets back to the target within a range of 10 consecutive frames, the tracking is resumed. Inspired by such a work, we evaluate the GSR $\uparrow$ with several different temporal ranges to detect a failure, specifically 1 frame (0.03s), 7 frames (0.25s), 15 frames (0.5s), 22 frames (0.75s), 30 frames (1s), 60 frames (2s), and 90 frames (3s).

Finally, we assessed the computational efficiency of the trackers. This has been done by quantifying the time difference (in seconds) between the time stamp associated with each frame and the time instant on which the localization for the respective frame is given by an algorithm. Considering that sports performance analysis requires the processing of all the frames for a smooth and continuous understanding, a tracker that is slow will accumulate time while processing all the frames and delay its predictions. Thus, it becomes interesting to know how much time should be waited in order to obtain the localization, and how such delay grows during the online processing of the video. We give such a measurement in seconds with Delay $\downarrow$.

C.1. Tracking Impact

As stated in the main paper, the output of tracking is of paramount importance for many high-level modules that produce fine-grained skiing performance analyses [29, 52, 53, 78, 80]. Thus we evaluated the trackers based on the impact they have on the accuracy of such solutions. We think that the development of effective tracking methodologies should be driven not only by tracking-specific results but also by the contribution the algorithms bring in improving the accuracy of the overall system.

As an exemplar high-level skiing performance under-

Table 9. **Overall and per-discipline results in the single-camera (SC) setting.** The F-Score ↑, Pr ↑, and Re ↑ scores are presented for each studied algorithm. This setting is easier to tackle by all the algorithms in general. The different skiing discipline pose challenges to the trackers in the same way as in the multi-camera (MC) setting.

Discipline	KCF	MOSSE	FEAR	SiamRPN++	GlobalTrack	LTMU	SeqTrack	KeepTrack	OSTrack	CoCoLoT	MixFormer	STARK	YOLO-SORT	STARK$_{FT}$	STARK$_{SKI}$
All	0.294	0.367	0.564	0.583	0.592	0.642	0.645	0.654	0.663	0.681	0.686	0.703	0.751 ③	0.836 ②	0.841 ①
	0.291	0.363	0.565	0.583	0.591	0.637	0.639	0.652	0.654	0.676	0.676	0.698	0.743	0.827	0.833
	0.299	0.376	0.572	0.592	0.601	0.651	0.681	0.664	0.704	0.694	0.658	0.717	0.763	0.854	0.858
AL	0.220	0.267	0.518	0.536	0.585	0.623	0.578	0.640	0.594	0.652	0.637	0.671	0.819	0.875	0.882
	0.218	0.265	0.524	0.542	0.595	0.622	0.576	0.641	0.590	0.650	0.634	0.672	0.814	0.876	0.886
	0.222	0.269	0.516	0.534	0.578	0.625	0.580	0.640	0.599	0.655	0.643	0.672	0.825	0.876	0.881
JP	0.389	0.487	0.641	0.663	0.677	0.702	0.747	0.705	0.763	0.738	0.765	0.761	0.855	0.899	0.907
	0.388	0.485	0.645	0.666	0.677	0.703	0.748	0.707	0.760	0.743	0.762	0.766	0.850	0.894	0.901
	0.390	0.489	0.640	0.660	0.678	0.701	0.747	0.703	0.767	0.734	0.770	0.758	0.863	0.907	0.914
FS	0.274	0.347	0.531	0.550	0.514	0.599	0.612	0.617	0.633	0.654	0.654	0.676	0.578	0.734	0.735
	0.268	0.338	0.525	0.540	0.502	0.586	0.595	0.607	0.612	0.636	0.633	0.656	0.566	0.711	0.713
	0.285	0.370	0.560	0.582	0.548	0.627	0.647	0.647	0.676	0.692	0.698	0.721	0.601	0.780	0.780

standing tasks to evaluate tracker's impact, we focused on the problem of 2D pose estimation of skier body and equipment [1, 53]. This task serves to obtain information regarding the position and orientation of specific human joints during exercises, and such an output is additionally exploited by even more high-level performance understanding modules such as 3D pose estimation [1, 79]. To estimate the image-level coordinates of a set of key-points that localize different parts of a skier's body (e.g. head, shoulders, hips, feet, etc.) and of particular points of interest of the skier's equipment (e.g. ski tips or tails), the available solutions [1, 53] first run an object detector [66, 67] to compute bounding-boxes for the athlete present in the input RGB image, and then crop image patches from such boxes that are successively given as input to a state-of-the-art deep neural network architecture (e.g. AlphaPose [32]) that predicts the key-point coordinates. Such a pose estimation network is trained by fine-tuning on ground-truth poses by exploiting input image patches extracted with bounding-boxes defined by the coordinates of the annotated key-points.

The aforementioned studies [1, 53] propose datasets of videos (with dedicated training and test sets) whose frames are sparsely labeled with the poses of body and equipment. The authors evaluate the proposed pipelines on such benchmarks but treat each frame as an independent image, and so during testing the object detector is run on every image before the pose estimation network. Considering the presence of videos, we use such datasets as a base for the evaluation of trackers as athlete localizers executed before the pose estimation step. Thus, we determine the tracker's impact by evaluating the accuracy of the pose estimation model, where the input of the latter is influenced by the output of the former. After having trained an AlphaPose instance [32] on the original training images [1, 53], we evaluate its accuracy on the test frames by inputting it with a patch extracted from a tracker's box prediction. The evaluation of the pose estimator is done through: the Percentage of Correct Keypoints (PCK ↑) which measures the number of predicted key-points, normalized by the number of all key-points [1], having a pixel distance lower than the 50% of the ground-truth-based head-neck distance; and the Mean Per Joint Position Error (MPJPE ↓) which measures the normalized pixel distance between predicted and corresponding ground-truth key-points [1]. The tracker's bounding-boxes are obtained by implementing the OPE protocol on the sequence of frames in between the first and the last pose annotation occurrences that refer to the same athlete. Indeed, we obtain boxes's top-left and bottom-right vertices by considering the lowest and greatest values in the key-points coordinates. The first bounding-box is considered for tracker initialization, while the others are for prediction evaluation. We respect the original training-test separations [1, 53]. For testing alpine skiing (AL) pose estimation on the Ski2DPose dataset, we used 11 video clips related to the 150 pose annotated images, while for pose estimation in ski jumping (JP) on the YouTube Skijump dataset we used 19 videos referring to the 118 annotated test images.

For the implementation and training of the AlphaPose instance [32], we employed the Alphapose v0.6 framework.[5] Specifically, we conducted two separate fine-tuning based on the ResNet50 model for Ski2DPose and YouTube Skijump. Both training sessions run for 250 epochs, employing a batch size of 32 and a learning rate of 0.001 decreased by a 0.1 factor every 70 epochs. During both training and testing, in the computation of the input image crop, a padding of 20% was added to the dimensions of the available bounding-box.

[5] https://github.com/MVIG-SJTU/AlphaPose

C.2. Implementation Details

All the code used for our study was implemented in Python and run on a machine with an Intel Xeon E5-2690 v4 @ 2.60GHz CPU, 320 GB of RAM, and 8 NVIDIA TITAN V GPUs.

D. Additional Results

This section reports the results of additional experiments we conducted.

Table 9 presents the performance achieved by the selected trackers in the case of skier tracking on SC videos. This setting is more similar to the problem of short-term visual object tracking [54,81] where the duration of the videos is shorter and they are captured by the same video camera (no camera shot-cuts are present). Application-wise, the conditions of SC tracking align with: the broadcasting requirements of skiing performance replay where just a specific section of the skiing performance is captured and played again; training processes where a trainer captures a specific section of the ski track/course with a smartphone for later video analysis. From the table, we observe that tracking a skier without camera-shots results easier in general. Generic object trackers show a larger improvement by tracking on SC videos than on MC ones. However, their tracking accuracy still remains lower than the skier-specific methods. Regarding the skiing disciplines, we notice that FS videos still cause the major drop in the overall performance.

D.1. Videos

Videos showing qualitative results of $\text{STARK}_{\text{SKI}}$ on SkiTB are available at https://machinelearning.uniud.it/datasets/skitb.