

Supplementary Material for “AdaSpot: Spend Resolution Where It Matters for Precise Event Spotting”

Supplementary Material

In this supplementary material, we provide additional details and analyses complementing the main paper. We first describe the datasets and evaluation protocols in more depth (Sec. A). Next, we expand on the implementation details of AdaSpot and the state-of-the-art methods (Sec. B). Sec. C presents extended ablation studies, while Sec. D and Sec. E provide further discussions on efficiency and randomness analyses, respectively. In Sec. F, we investigate the sensitivity of PES methods to the choice of post-processing. Finally, Sec. G reports additional per-class and qualitative results for AdaSpot.

A. Data and evaluation protocols description

In this section, we first provide additional details about the datasets, followed by further clarification of the evaluation protocols.

A.1. Datasets description

We evaluated AdaSpot on five datasets: Tennis [25], FineDiving [24], FineGym [17], and F3Set [12] under the Precise Event Spotting (PES) setting, as well as SoccerNet Ball Action Spotting (SN-BAS) [8, 18] under the less strict Event Spotting (ES) setting, which requires lower temporal precision. In the following, we provide additional details for each dataset.

Tennis. The Tennis dataset, originally introduced in Zhang et al. [25] and later extended by Hong et al. [9] consists of 3 345 video clips, each corresponding to a single tennis point, extracted from 28 matches. The videos have frame rates ranging from 25 to 30 frames per second. In total, the dataset contains 33 791 precisely annotated events across six classes –“serve”, “swing”, and “ball bounce”, each distinguished between near- and far-court–, with the class-wise distribution provided in Tab. 5. All annotated events are therefore ball-centric, indicating that the regions of interest in this dataset are predominantly around the ball’s spatial location.

FineDiving. The FineDiving dataset, introduced by Xu et al. [24], comprises 3 000 diving clips recorded at 25

frames per second. In total, it contains 7 010 events corresponding to transitions into somersaults, categorized into four classes –“pike”, “tuck”, “twist”, and “entry”–, with per-class frequencies provided in Tab. 6. All annotated events involve a single primary athlete, so the regions of interest are naturally centered on that athlete.

FineGym. The FineGym dataset, introduced by Shao et al. [17], comprises 5 374 untrimmed videos of gymnastics performances, originally recorded at frame rates between 25 and 60 frames per second. Following Hong et al. [9], the videos’ frame rates are standardized to 25-30 fps. In total, the dataset contains 80 166 events corresponding to the start and end of various gymnastics actions –such as “floor exercise turns”, “uneven bars dismounts”, and “balance beam turns”–, spanning 32 event classes, with per-class frequencies summarized in Tab. 7. As these events are athlete-centric, the regions of interest are predominantly focused on the main athlete.

F3Set. The F3Set dataset, introduced by Liu et al. [12] contains 11 584 video clips, each corresponding to a tennis point, extracted from 114 matches featuring 75 players. Videos have frame rates between 25 and 30 fps. In total, the dataset includes 42 846 precisely annotated events across 365 classes, each representing a combination of categories such as the player hitting the ball, court location, body side, shot type, shot direction, shot technique, player movement, and shot outcome. While F3Set is similar to Tennis, it features far more fine-grained events. We omit per-class statistics and analyses due to the large number of classes. As in Tennis, all events are ball-centric.

SN-BAS. The SN-BAS dataset [8, 18] consists of untrimmed videos from seven English Football League matches recorded at 25 fps. In total, the dataset contains 12 422 annotated ball-related events. The event classes include: “pass”, “drive”, “header”, “high pass”, “out”, “cross”, “throw-in”, “shot”, “ball-player block”, “player successful tackle”, “free-kick”, and “goal”, with per-class frequencies listed in Tab. 8. As the events are ball-centric, the relevant regions of interest are naturally centered around the ball.

A.2. Evaluation protocols

For the Tennis, FineDiving, FineGym, and F3Set datasets under the PES setting, we follow the evaluation protocol proposed by Hong et al. [9], using the same training, validation, and test splits. The task is evaluated using mean Average Precision at a given temporal tolerance δ , denoted as $mAP@{\delta}$. For these datasets, we report results using temporal tolerances of $\delta \in \{0, 1, 2\}$ frames.

For SN-BAS, which has primarily been used for challenge purposes [4, 5], many existing methods rely on dataset-specific tricks, alternative data splits, or external data sources. To ensure fair and reproducible benchmarking, we introduce a standardized evaluation protocol. We adopt the original data splits, training on the four-game training set, using the one-game validation for early stopping, and reporting results on the two-game test set, while discarding the two-game challenge set with hidden ground-truth. Following [7], we exclude the “free-kick” and “goal” event classes due to their extremely low frequency—six and two examples, respectively, in the test split—which makes the metric highly sensitive to single correct or incorrect predictions. To maintain a more stable and meaningful evaluation, we remove these classes from our analysis. For this dataset, under the less strict ES setting, the task is evaluated using mean Average Precision with temporal tolerances of $\delta \in \{0.5, 1\}$ seconds.

B. Implementation details

To ensure reproducibility, in this section we provide implementation details for AdaSpot, as well as those for the state-of-the-art models used in our comparisons. We also describe the adaptations required to apply redundancy-aware methods to the PES setting.

B.1. AdaSpot

In addition to the implementation details provided in Sec. 4.1, we train AdaSpot on clips of $L = 100$ frames with a batch size of 4. For the two model variants, AdaSpot^s, and AdaSpot^b, which use different feature-extractor sizes, the hidden dimensions are set to $d = 368$ and $d = 608$, respectively. The RoI selector uses an upsampling factor of $k = 8$, and the threshold parameter τ is empirically tuned for each dataset. To mitigate class imbalance, the cross-entropy-loss assigns a weight of $w = 5$ to positive classes, and the loss coefficients are set to $\lambda_f = \lambda_l = \lambda_h = \frac{1}{3}$. For F3Set, we use per-event-category prediction heads to handle multiple categories, with each event class probability computed as the product of its category probabilities. Each epoch consists of 5 000 randomly sampled clips. We train the models for 25 epochs on FineDiving, 50 epochs on Tennis and SN-BAS, and 100 epochs on the larger FineGym and F3Set datasets. Optimization is performed with AdamW [13], us-

ing a base learning rate of $8e-4$, five warm-up epochs, and cosine learning-rate decay. Soft Non-Maximum Suppression uses a window size of 2 frames for PES and 12 for ES. The method is implemented in PyTorch [15], and all models are trained on a single NVIDIA RTX 6000 Ada Generation GPU.

B.2. State-of-the-art models

PES setting. In the PES setting, we compare AdaSpot against several state-of-the-art methods. Specifically, we include E2E-Spot [9] in two variants—E2E-Spot_{200MF} and E2E-Spot_{800MF}—which use RegNetY-200MF and RegNetY-800MF as feature extractors, respectively. We also include UGLF [19] and T-DEED [22], the latter likewise evaluated in two configurations, T-DEED_{200MF} and T-DEED_{800MF}, based on the same RegNetY backbones. In addition, we report results for Santra et al. [16], and F³ED for F3Set. We exclude alternative approaches considered in Hong et al. [9], such as two-stage methods with pre-extracted features, due to their lower performance, focusing the comparison on end-to-end models that achieve high performance. As discussed in Sec. F, the choice of post-processing technique can notably affect the evaluation metrics. To ensure a fair comparison, we report results for all methods using the same postprocessing procedure specified in Sec. C. For E2E-Spot and T-DEED, which originally report results with different postprocessing strategies, we run inference using their publicly available checkpoints and update the postprocessing accordingly. Santra et al. [16] already reports results using Soft-NMS with a window of 2. For UGLF, no public checkpoints are available; thus, we report the results as provided in their original paper, which uses a different postprocessing setup. Finally, for F³ED, we run inference using the publicly available checkpoints and modify the code to compute the mAP metrics not included in the original implementation. We additionally compare AdaSpot and F³ED under their native evaluation metrics in Sec. G.

ES setting. In the ES setting, we compare AdaSpot against E2E-Spot (in two variants: E2E-Spot_{200MF} and E2E-Spot_{800MF}), T-DEED (T-DEED_{200MF} and T-DEED_{800MF}), and Santra et al. [16]. Since none of these methods provide pre-trained models on SN-BAS under the evaluation protocol specified in Sec. A.2, we re-implemented them under our training pipeline. For E2E-Spot and T-DEED, we leverage their publicly available code. For T-DEED’s SGP-Mixer module, we adopt $B = 2$ layers, a kernel size of $ks = 9$, and a scalable factor of $r = 4$, consistent with their SN-BAS experiments. For Santra et al. [16], no public code is available, so we implemented their proposed AS-TRM module from scratch. All methods are trained on sequences of $L = 100$ frames with a spatial resolution of 398×224 .

B.3. Redundancy-aware methods

We provide additional implementation details for the comparison of AdaSpot with alternative redundancy-aware approaches in Sec. 4.3 of the main paper. We first report results for AdaSpot under three configurations with low-resolution inputs of $(W_l, H_l) = \frac{1}{4}(W_h, H_h)$, $(W_l, H_l) = \frac{3}{8}(W_h, H_h)$, and $(W_l, H_l) = \frac{1}{2}(W_h, H_h)$, as well as for a single low-resolution baseline that uses only the low-resolution branch under the same input resolutions. For the redundancy-aware methods, we adopt the taxonomy shown in Fig. 1, which distinguishes architecture-based methods –those that mitigate redundancy at the feature level– from input-based methods –those that address redundancy at the input level. Since AdaSpot targets spatial redundancy, which is more relevant for the PES task, we restrict our comparisons to methods that explicitly handle spatial redundancy. Specifically, we evaluate AdaSpot against: (i) deformable convolutions [6], applied spatially; (ii) sparse convolutions [10], also applied spatially in two variants –one using saliency maps (Sparse-Saliency) and one using learned gating mechanisms (Sparse-Learned) to select sparsity locations; (iii) learnable pixel-space cropping (AdaFocus-v2 [20]); (iv) learnable feature-space cropping with variable size regions (Uni-AdaFocus [21]); and (v) saliency-driven frame warping [11]. Additional details for each approach are provided below.

Deformable convolutions. For this approach, we adopt a simplified version of the AdaSpot architecture consisting of a single branch that processes frames at a fixed spatial resolution. Concretely, we retain one feature extractor, the temporal modeler, and the prediction head, while removing the RoI selector, the second feature extractor, the linear projectors, and the aggregation module. We then incorporate deformable convolutions into the remaining feature extractor. Specifically, all convolutions outside the initial “stem” block –so as to preserve standard dense early processing– with kernel size larger than 1×1 are replaced by deformable convolutions with matching configuration. We report results for two variations of this approach, corresponding to input spatial resolutions of 398×224 and 796×448 , which yield different computational costs.

Sparse-Saliency. This approach uses the same simplified architecture as the deformable-convolution baseline, but replaces the designated dense convolutions with sparse convolutions instead. The sparse activation pattern is determined using saliency maps: for each convolution, we compute a saliency map by channel-wise averaging the input features, following the procedure used in AdaSpot. We then retain the top 25% of positions within each frame’s feature maps with the highest activations as the active sparse locations. As in the deformable convolutions approach, we report results for two variants with input spatial resolutions

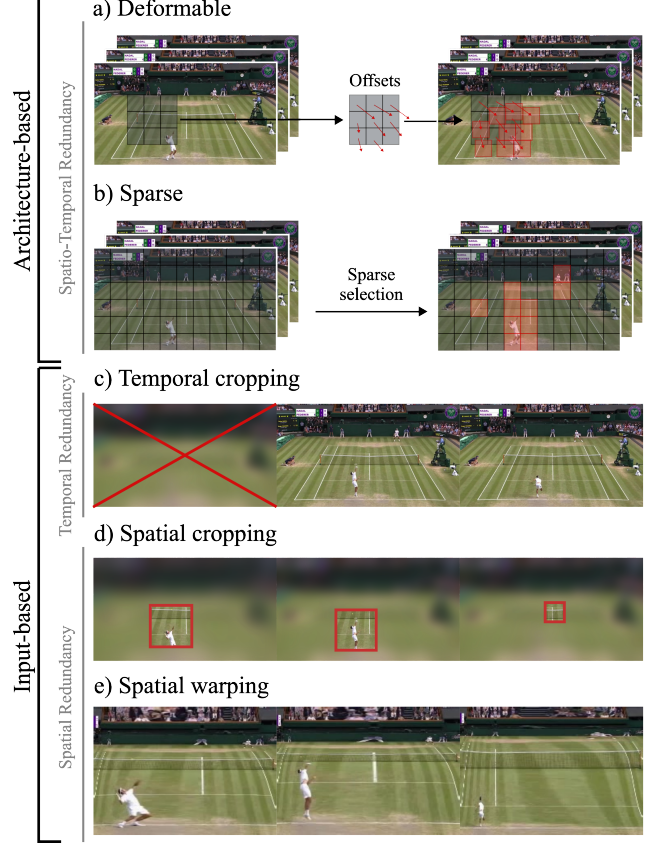


Figure 1. Illustration of the taxonomy of methods addressing spatio-temporal redundancy. We categorize approaches into *architecture-based* and *input-based*, and indicate whether each method handles spatial redundancy, temporal redundancy, or both.

of 398×224 and 796×448 .

Sparse-Learned. This approach is similar to Sparse-Saliency, but the sparse activation pattern is learned end-to-end using a lightweight gating module that predicts per-position importance scores via a linear layer followed by a sigmoid. We then select the top 25% positions using a hard top-k during the forward pass, while employing a straight-through estimator (STE) [2] in backpropagation to allow gradients to flow through the soft scores. The resulting masked features are then processed by the convolutional layer. As in the other methods, we evaluate two variants with input spatial resolutions of 398×224 and 796×448 .

AdaFocus-v2. For this approach, we adopt the same architecture as AdaSpot, replacing the RoI selector with the one proposed in Wang et al. [20]. Specifically, their RoI selector takes the feature maps F_s as input and processes them through a series of spatial and temporal modules to produce per-frame predictions indicating the center of the region to crop. The approach is made differentiable via their learnable cropping mechanism, which incorporates a stop-gradient operation to improve training stability. We eval-

uate three variants of this method, corresponding to low-resolution inputs of $(W_l, H_l) = \frac{1}{4}(W_h, H_h)$, $(W_l, H_l) = \frac{3}{8}(W_h, H_h)$, and $(W_l, H_l) = \frac{1}{2}(W_h, H_h)$.

Uni-AdaFocus. This approach is analogous to the previous one, but replaces the RoI selector with the version proposed in Wang et al. [21]. Specifically, their method learns crop positions in the feature-space to improve training stability and is adapted to allow variable-size regions. As with the previous baseline, we evaluate three variants with low-resolution inputs of $(W_l, H_l) = \frac{1}{4}(W_h, H_h)$, $(W_l, H_l) = \frac{3}{8}(W_h, H_h)$, and $(W_l, H_l) = \frac{1}{2}(W_h, H_h)$.

Saliency warping. This approach uses the same AdaSpot architecture, but replaces the selected regions in the high-resolution branch with warped frames that emphasize the relevant regions, following Liu et al. [11]. We use the same saliency maps extracted for AdaSpot to guide the warping, as they provide reliable estimates of important regions. The warped frames are generated using the method proposed in Liu et al. [11]. As with other baselines, we evaluate three variants with low-resolution inputs of $(W_l, H_l) = \frac{1}{4}(W_h, H_h)$, $(W_l, H_l) = \frac{3}{8}(W_h, H_h)$, and $(W_l, H_l) = \frac{1}{2}(W_h, H_h)$.

C. Additional ablation studies

In this section, we extend the ablation analysis presented in Sec. 4.3 of the main paper. Specifically, we first provide a more detailed examination of the components and parameters of our proposed AdaSpot approach. We then analyze the instability of AdaSpot compared to learnable-based alternatives, and finally, we further examine and discuss the selected RoIs for AdaSpot in comparison with those of alternative redundancy-aware methods that operate in the input space.

C.1. Extended component analysis

We extend the component analysis from Sec. 4.3 by first providing a visual examination of the *center bias* issue that arises when using zero-padding. We then report additional ablations on key components and parameters of AdaSpot, including alternative fusion strategies, different crop sizes, weight-sharing between the feature extractors of the low- and high-resolution branches, employing adaptive RoI aspect ratios, selecting multiple RoIs per frame, and analyzing the sensitivity of the τ parameter.

Center bias extended analysis. In Sec. 4.3 of the main paper, we reported a performance drop when replacing replicate padding with zero-padding. We attribute this drop to a *center bias* introduced by zero-padding, which artificially reduces activation strength near the image borders [1]. Fig. 2 provides additional qualitative evidence for this effect by visualizing the resulting saliency maps and the RoIs selected when zero-padding is used. Although the saliency

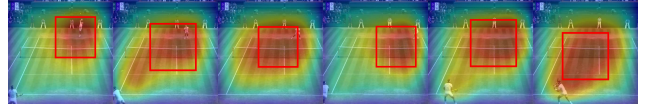


Figure 2. Qualitative visualization of saliency maps and the resulting selected RoIs when zero-padding is applied on the Tennis dataset. With zero-padding, the RoI selector ends up biased towards the central part of the frames.

Table 1. Extended ablation study of AdaSpot components on Tennis and SN-BAS, evaluating the impact of alternative fusion mechanisms, crop sizes, backbone reuse for both low- and high-resolution branches, adaptive RoI aspect ratios, and multiple RoIs per frame.

Experiment	Tennis			SN-BAS	
	$\delta = 0f$	1f	2f	$\delta = 0.5s$	1s
AdaSpot	73.30	96.90	97.47	53.02	56.43
(a) <i>Fusion mechanism</i>					
mean	71.26	96.48	97.07	50.55	54.56
product	71.88	96.75	97.37	51.24	54.95
linear	71.36	96.37	96.96	51.30	55.39
frame-gated	71.86	96.27	96.80	52.03	56.25
channel-gated	72.93	96.87	97.41	52.12	55.60
(b) <i>Crop size</i>					
56×56	71.05	96.52	97.18	51.53	55.61
84×84	72.19	96.66	97.28	51.16	55.26
168×168	72.45	96.89	97.47	52.08	55.58
224×224	73.02	96.77	97.28	50.60	54.66
(c) <i>Extractor reuse</i>					
yes	71.70	96.52	97.19	51.90	56.04
(d) <i>RoI aspect ratio</i>					
adaptive	71.66	96.62	97.24	51.61	54.97
(e) <i># RoIs per frame</i>					
2 RoIs per frame	72.06	96.83	97.35	49.19	52.70

maps generally track the ball, activations near the borders are notably weaker than those at the center, leading the RoI selector to avoid choosing regions along the frame boundaries—even when the ball is located there. Consequently, the high-resolution branch receives less semantically meaningful crops, which negatively affects performance. This issue is most pronounced on the Tennis dataset and appears in certain runs, but when it arises, it can substantially degrade AdaSpot’s effectiveness. In contrast, we do not observe any such behavior when using replicate padding, across all experiments and random seeds.

Additional component and parameter ablations for AdaSpot. In Tab. 1(a) we compare AdaSpot’s max-based fusion of F_l' and F_h' with several **alternative fusion mechanisms**. Specifically, we evaluate: (i) *mean*—the per-position element-wise average; (ii) *product*—the element-wise (Hadamard) product; (iii) *linear*—concatenating the feature vectors along the channel dimension and projecting back to dimension d through a linear layer; (iv) *frame-gated*—a per-frame gating mechanism that predicts a scalar α from the features and fuses them as $F_f = \alpha F_l' + (1 - \alpha) F_h'$; and

(v) *channel-gated* –the same gating approach but predicting channel-wise gates instead of a single scalar. As shown by the results, none of these alternatives surpass the simple max-based aggregation, with several of them also incurring additional computational overhead. Tab. 1(b) reports results for **varying crop sizes**. Reducing the crop below 112×112 slightly decreases performance, likely because smaller regions either capture less content or are downsampled to lower resolution when resized to (W_r, H_r) . Increasing the crop size yields results closer to the baseline but does not surpass it, which we attribute to larger RoIs introducing extra context that is not task-relevant. Tab. 1(c) shows that **reusing extractor parameters** for both the low- and high-resolution branches still achieves strong performance with only minor drops. This shows that AdaSpot can be made more parameter-efficient, reducing total parameters by 37% while decreasing the strictest metrics by only -1.60 and -1.12 on Tennis and SN-BAS, respectively. Additionally, Tab. 1(d) evaluates **adaptive RoI aspect ratios**, where the RoI is no longer constrained to a fixed aspect ratio. Instead, here we take the rectangular region according to the saliency spread without enforcing this constraint. This modification results in a performance drop of -1.64 and -1.41 on Tennis and SN-BAS, respectively. We attribute this to the increased complexity of modeling RoIs with varying aspect ratios, which complicates training and reduces performance. In Tab. 1(e) we evaluate using **multiple RoIs per frame**. We extend AdaSpot to the multi-RoI setting by selecting a second region corresponding to the highest remaining saliency after excluding the first RoI for each frame. This results in two RoI clips that are processed through the shared high-resolution extractor and aggregated using element-wise maximum. For simplicity, a fixed region size is used in these experiments. The results show that incorporating more than one RoI consistently degrades performance, indicating that additional regions do not provide complementary information and instead introduce noise. This finding aligns with our qualitative analysis (Sec. 4.4), where saliency maps typically highlight a single dominant region, suggesting that a single RoI suffices for current PES benchmarks. While multi-RoI modeling could benefit scenarios with multiple simultaneous events, such dynamics are not present in the standard PES datasets. A more extensive study of multi-RoI extensions of AdaSpot, evaluated on datasets with concurrent events and multiple relevant regions, is therefore left for future work. Finally, Fig. 3 analyzes the sensitivity to the **threshold parameter** τ . Both datasets exhibit similar trends, with two main performance peaks, at $\tau = 0$ and around $\tau = 0.3$. The peak at $\tau = 0$ arises from using fixed-size RoIs, which simplifies modeling in the high-resolution branch despite occasionally omitting context. As τ increases, performance initially decreases, indicating that the added contextual information

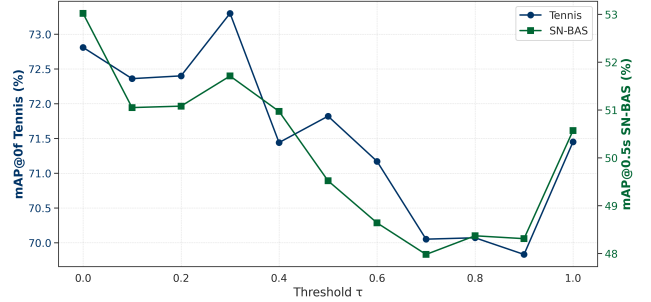


Figure 3. Sensitivity of AdaSpot performance (mAP@0f / mAP@0.5s) to variations in the threshold parameter τ . The blue line denotes the Tennis dataset (left y-axis), while the green line denotes the SN-BAS dataset (right y-axis).

Table 2. Comparison of instability under the strictest metric for different learnable cropping methods with AdaSpot. Bold indicates best; (mean \pm std. across 3 runs).

Dataset	Low & high-res features			High-res features only		
	AF-v2	Uni-AF	AdaSpot	AF-v2	Uni-AF	AdaSpot
Tennis	70.6 \pm 1.5	70.2 \pm 1.0	73.3 \pm 0.5	68.8 \pm 0.9	65.9 \pm 0.9	71.9 \pm 0.4
SN-BAS	49.0 \pm 2.0	49.6 \pm 1.3	53.0 \pm 0.5	23.1 \pm 15.1	39.4 \pm 3.3	52.1 \pm 0.7

does not compensate for the difficulty of modeling variable-size RoIs. Around $\tau = 0.3$, the added context becomes beneficial enough to counteract this effect, producing the second peak. Beyond this range, performance declines once RoIs include excessive non-informative content while still varying in size. At $\tau = 1$, we observe a small final peak, as the RoI becomes the full frame, again yielding fixed-size regions that simplify modeling –though at the cost of negating the purpose of the high-resolution branch, which now processes downsampled full-view clips.

C.2. Instability analysis of learnable cropping

Tab. 2 presents a comparative instability analysis of AdaSpot against alternative learnable cropping methods: AdaFocus-v2 (AF-v2) and Uni-AdaFocus (Uni-AF). Across datasets, AdaSpot consistently achieves lower standard deviation, indicating more stable training. AF-v2 exhibits high variability, which is partially mitigated in Uni-AF. In addition, AdaSpot demonstrates more robust RoI selection, as reflected by higher performance when using high-resolution features only. In contrast, AF-v2 and Uni-AF produce more failure cases. These results highlight AdaSpot’s improved training stability and RoI robustness.

C.3. Qualitative RoI comparison

Fig. 4 compares the RoIs selected by our AdaSpot with those produced by input-based alternatives, specifically AdaFocus-v2 and Uni-AdaFocus. As shown, these alternative methods frequently fail to capture task-relevant regions, introducing noise during training and diminishing the effectiveness of the high-resolution branch –ultimately leading to

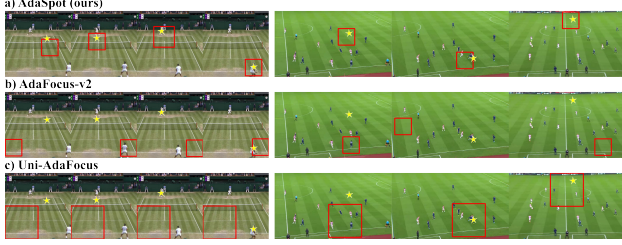


Figure 4. Qualitative comparison of the RoIs selected by AdaSpot, AdaFocus-v2, and Uni-AdaFocus on the Tennis (left) and SN-BAS (right) datasets. For visualization, we mark the ball position in each frame with a star, as actions in these datasets occur around the ball; thus, relevant RoIs should contain or closely surround it.

the performance drops reported in Sec. 4.3 of the main paper. On Tennis, both AdaFocus-v2 and Uni-AdaFocus tend to converge to largely static corner crops, likely because some actions commonly occur near those areas. On SN-BAS, the crops move more dynamically, and Uni-AdaFocus localizes relevant regions more reliably (*e.g.*, around the ball). However, its adaptive region size often saturates to the maximum allowed area, causing fine-grained details to be lost after resizing to (W_r, H_r) . In contrast, AdaSpot consistently selects stable, semantically meaningful RoIs. As discussed in the main paper, we attribute the limitations of such learnable-cropping approaches to the training instabilities identified in prior work [21], which our training-free RoI selector inherently avoids.

D. Efficiency analysis

In this section, we extend the efficiency analysis presented in Sec. 4.2 of the main paper. Tab. 3 reports the number of parameters and GFLOPs required to process a single clip under both the PES setting (base resolution 224×224) and the ES setting (base resolution 398×224). For Santra et al. [16], these values are derived from our re-implementation of their ASTRM module, which may therefore differ slightly from those originally reported. We exclude UGLF [19] from the comparison due to missing details regarding their vision-language module in the released code. While E2E-Spot_{200MF}, T-DEED_{200MF}, and Santra et al. [16] all employ RegNetY-200MF as their base extractor, E2E-Spot stands out among the most efficient in both parameter count and GFLOPs. T-DEED exhibits comparable computational cost but is substantially more parameter-intensive due to its SGP-Mixer module used for temporal modeling. In contrast, Santra et al. [16] maintains a low parameter count but incurs higher GFLOPs because the ASTRM module is inserted early in the backbone, where feature maps are still high resolution, thereby increasing the overall computational cost. AdaSpot^s, which uses the same base extractor, introduces only a marginal increase in parameters and GFLOPs relative to E2E-Spot_{200MF}. The additional pa-

Table 3. Efficiency comparison of AdaSpot with state-of-the-art methods in both the PES setting (typically using 224×224 inputs) and the ES setting (using 398×224 inputs). For each configuration, we report the number of parameters (in millions) and the computational cost in GFLOPs.

Model	PES		ES	
	P(M)	GFLOPs	P(M)	GFLOPs
E2E-Spot _{200MF} [9]	4.49	23.13	4.49	40.78
E2E-Spot _{800MF} [9]	12.70	84.93	12.70	150.02
T-DEED _{200MF} [23]	16.42	21.97	12.31	39.58
T-DEED _{800MF} [23]	64.26	86.34	46.22	151.31
Santra et al. [16]	6.46	57.84	6.84	82.51
AdaSpot^s	7.58	29.78	7.58	46.18
AdaSpot^b	10.63	36.78	10.63	90.04

rameters arise from duplicating the extractor for the high-resolution branch, while the extra computation stems from processing the RoI clips through this branch—adding approximately 6 GFLOPs for our standard 112×112 RoI configuration. This small overhead enables AdaSpot to preserve fine-grained details and yields substantial performance improvements (see Sec. 4.2 of the main paper), resulting in a stronger efficiency-accuracy trade-off. When comparing larger extractor configurations—E2E-Spot_{800MF}, T-DEED_{800MF}, and AdaSpot^b—we observe that AdaSpot^b, despite using a smaller backbone (RegNetY-400MF) and thus being more efficient, still achieves state-of-the-art performance across both PES and ES datasets (see Sec. 4.2 of the main paper). Additionally, for AdaSpot, inference on a single clip requires only 1.97GB of GPU memory, enabling inference even on small GPUs.

E. Randomness analysis

Training deep neural networks involves multiple sources of randomness (*e.g.*, data sampling, weight initialization, and data augmentation), which can lead to noticeable performance variability across runs. Despite this, most PES methods report results from a single training run, due to the substantial computational cost of these pipelines. This practice can produce benchmarks that are sensitive to run-to-run fluctuations, making claims difficult to verify or reproduce. To provide more robust evaluations, we report results over three runs using different random seeds. For all experiments, we report the mean performance, and for the main results, we additionally provide the standard deviation to reflect variability across runs. While more runs would allow more rigorous statistical analysis—three runs are insufficient for reliable significance testing—the high computational demands of PES frameworks make extensive multi-run evaluation impractical. Nevertheless, our three-run reporting offers improved robustness over the single-run convention used in prior work.

Table 4. Post-processing sensitivity analysis on the Tennis dataset. We report results for standard NMS and Soft-NMS using different window sizes ω . Bold and underlined values indicate the best and second-best results.

		Tennis		
Post-processing		$\delta = 0f$	1f	2f
NMS [14]	$\omega = 1$	62.82	96.93	<u>97.61</u>
	$\omega = 2$	62.45	96.61	97.64
	$\omega = 3$	62.35	96.11	97.58
	$\omega = 4$	62.32	95.94	97.47
	$\omega = 5$	62.29	95.84	97.29
Soft-NMS [3]	$\omega = 1$	75.05	96.02	96.50
	$\omega = 2$	<u>73.30</u>	96.90	97.47
	$\omega = 3$	71.53	<u>96.92</u>	97.56
	$\omega = 4$	70.35	96.81	97.60
	$\omega = 5$	69.36	96.70	97.59

F. Post-processing analysis

In this section, we analyze the sensitivity of PES methods to the choice of post-processing. Tab. 4 presents AdaSpot results on the Tennis dataset under different post-processing configurations. Specifically, we compare standard Non-Maximum Suppression (NMS) [14], and Soft Non-Maximum Suppression (Soft-NMS) [3], evaluating multiple window sizes $\omega \in \{1, 2, 3, 4, 5\}$. As shown, Soft-NMS generally outperforms NMS for the strictest metric (mAP@0f) while achieving comparable results for looser metrics. Within Soft-NMS, smaller window sizes slightly favor stricter metrics, whereas larger windows benefit more relaxed metrics. Following prior work [16, 22], we adopt a configuration that balances performance across all tolerances and use Soft-NMS with $\omega = 2$. For ES experiments, where more relaxed evaluation protocols are used, larger window sizes are preferable; in this case, we find $\omega = 12$ to offer the best trade-off. To ensure fair comparisons with state-of-the-art methods (Sec. 4.2 main paper), whenever possible, all results are re-extracted using the same post-processing settings.

G. Additional results and visualizations

In this section, we first analyze the per-class performance of AdaSpot in comparison with other state-of-the-art methods (E2E-Spot and T-DEED), and provide an approximate per-class evaluation of the RoI selection. We then present additional results, including F3Set evaluations under their proposed metrics, as well as visualizations of the generated saliency maps, the selected RoIs, and the corresponding model predictions for AdaSpot.

Table 5. Per-class analysis on the Tennis dataset. For each event class, we report the total number of observations and the AP@0f results for the best-performing versions of E2E-Spot, T-DEED, and AdaSpot, as well as for an AdaSpot variant using high-resolution features only (HR-only). Event classes are sorted in descending order of observations. The best result per class is highlighted in bold, and the second-best is underlined.

Event	N° observations	AP ($\delta = 0f$)			
		E2E-Spot	T-DEED	AdaSpot	HR-only
Far-court ball bounce	8150	76.91	59.47	77.20	75.38
Near-court ball bounce	8127	<u>76.79</u>	68.19	78.98	78.91
Far-court swing	7123	<u>53.24</u>	41.52	64.76	58.77
Near-court swing	7044	<u>56.42</u>	48.85	58.83	58.49
Near-court serve	1690	<u>76.79</u>	67.91	79.24	78.67
Far-court serve	1657	<u>80.10</u>	64.66	85.09	81.95

Table 6. Per-class analysis on the FineDiving dataset. For each event class, we report the total number of observations and the AP@0f results for the best-performing versions of E2E-Spot, T-DEED, and AdaSpot. Event classes are sorted in descending order of observations. The best result per class is highlighted in bold, and the second-best is underlined.

Event	N° observations	AP ($\delta = 0f$)		
		E2E-Spot	T-DEED	AdaSpot
Entry	2984	22.51	24.02	26.74
Som(s).Pike	2152	<u>27.21</u>	23.14	27.58
Som(s).Tuck	1071	<u>31.70</u>	21.72	32.23
Twist(s)	803	<u>18.60</u>	16.45	22.51

G.1. Per-class results

Here, we report per-class results of AdaSpot compared with E2E-Spot and T-DEED, using the best-performing version of each model for each dataset. On the **Tennis** (Tab. 5) and **FineDiving** (Tab. 6) datasets, we observe the same trend as in the aggregated results from the main paper, with AdaSpot outperforming the other two methods across all event classes. In Tennis, the most notable improvements over E2E-Spot occur on “far-court swings” and “far-court serves”, highlighting that AdaSpot is particularly effective for far-view actions where uniform resolution down-sampling can hinder performance. By focusing higher-resolution attention on relevant regions, AdaSpot better captures these challenging events. On **FineGym** (Tab. 7), the performance across methods is generally similar, but AdaSpot maintains competitive results across all classes, achieving strong overall performance. Finally, on **SN-BAS** (Tab. 8), AdaSpot again demonstrates superiority, achieving the best results for all but two classes. These results confirm that the improvements introduced by AdaSpot are consistent across most event categories, reinforcing its general effectiveness.

G.2. Per-class RoI analysis

Per-class RoI analysis is limited by the lack of ground-truth RoIs. However, evaluating an AdaSpot variant that

Table 7. Per-class analysis on the FineGym dataset. For each event class, we report the total number of observations and the AP@0f results for the best-performing versions of E2E-Spot, T-DEED, and AdaSpot. Event classes are sorted in descending order of observations. The best result per class is highlighted in bold, and the second-best is underlined.

Event	N° observations	AP ($\delta = 0f$)		
		E2E-Spot	T-DEED	AdaSpot
Uneven bars circles start	6612	11.32	10.26	<u>10.28</u>
Uneven bars circles end	6612	20.19	<u>19.89</u>	19.63
Balance beam leap_jump.hop start	4787	17.52	19.72	<u>18.07</u>
Balance beam leap_jump.hop end	4787	10.31	12.64	<u>10.81</u>
Balance beam flight_salto start	4187	19.84	<u>22.72</u>	24.35
Balance beam flight_salto end	4187	6.86	7.76	<u>7.48</u>
Uneven bars transition_flight start	3389	29.86	<u>29.60</u>	26.65
Uneven bars transition_flight end	3389	30.73	26.99	<u>28.09</u>
Floor exercise leap_jump.hop start	3238	27.32	<u>26.14</u>	25.43
Floor exercise leap_jump.hop end	3238	16.41	14.10	<u>14.91</u>
Floor exercise back_salto start	2978	35.88	33.26	<u>34.86</u>
Floor exercise back_salto end	2978	13.61	11.95	<u>12.83</u>
Balance beam flight_handspring start	2893	17.64	19.93	<u>19.08</u>
Balance beam flight_handspring end	2893	23.91	28.80	<u>26.50</u>
Vault (timestamp 0)	2031		1.90	<u>2.36</u>
Vault (timestamp 1)	2031	22.54	<u>22.53</u>	20.15
Vault (timestamp 2)	2031	35.28	<u>39.90</u>	41.80
Vault (timestamp 3)	2031	5.43	7.07	<u>6.29</u>
Uneven bars flight_same_bar start	1624	<u>27.30</u>	27.85	25.63
Uneven bars flight_same_bar end	1624	26.50	27.58	<u>26.82</u>
Balance beam turns start	1371	<u>12.47</u>	13.98	11.56
Balance beam turns end	1371	4.67	5.33	4.64
Floor exercise from_salto start	1345	26.48	<u>26.83</u>	29.60
Floor exercise from_salto end	1345	8.97	8.52	<u>8.70</u>
Uneven bars dismounts start	1227	34.37	<u>33.50</u>	33.03
Uneven bars dismounts end	1227	10.65	<u>8.55</u>	7.80
Balance beam dismounts start	1218	21.70	34.94	<u>27.86</u>
Balance beam dismounts end	1218	7.11	<u>6.89</u>	4.96
Floor exercise turns start	1103	9.07	12.53	<u>11.41</u>
Floor exercise turns end	1103	11.07	15.30	<u>13.52</u>
Floor exercise side_salto start	49	22.35	8.49	<u>19.80</u>
Floor exercise side_salto end	49	<u>2.86</u>	1.59	7.70

Table 8. Per-class analysis on the SN-BAS dataset. For each event class, we report the total number of observations and the AP@0.5s results for the best-performing versions of E2E-Spot, T-DEED, and AdaSpot, as well as for an AdaSpot variant using high-resolution features only (HR-only). Event classes are sorted in descending order of observations. The best result per class is highlighted in bold, and the second-best is underlined.

Event	N° observations	AP ($\delta = 0.5s$)			
		E2E-Spot	T-DEED	AdaSpot	HR-only
Pass	4985	<u>85.15</u>	83.44	85.94	85.11
Drive	4300	<u>81.55</u>	77.25	81.77	81.50
High pass	761	79.30	76.33	<u>78.68</u>	75.18
Header	713	<u>68.14</u>	54.27	68.87	62.62
Ball out of play	551	16.97	<u>19.79</u>	23.01	21.15
Throw-in	362	<u>67.74</u>	58.48	70.52	65.40
Cross	261	<u>64.27</u>	62.64	69.66	52.65
Ball player block	223	<u>24.94</u>	16.46	24.98	21.34
Shot	169	<u>51.23</u>	44.31	55.21	53.21
Player successful tackle	74	5.64	0.92	<u>3.77</u>	1.84

uses only high-resolution features provides an approximate measure of RoI precision for each event class. We report these values for Tennis and SN-BAS in Tab. 5 and Tab. 8. As shown in the tables, in Tennis, the largest performance drops compared to the full AdaSpot model occur for far-court events, indicating less precise RoI selection for distant actions. In contrast, close-court events show performance near the baseline, suggesting accurate RoI selection

Table 9. Comparison of AdaSpot with F³ED on the F3Set dataset using standard PES mAP metrics, as well as F1 and Edit scores. Results show the mean over three random seeds with the corresponding standard deviation (\pm). Bold and underlined values indicate the best and second-best results.

	mAP@0f	mAP@1f	mAP@2f	F1 _{evt}	Edit
F ³ ED [12]	24.8	60.7	64.8	40.3	74.0
AdaSpot ^a	<u>53.55\pm 1.2</u>	<u>67.76\pm 0.8</u>	<u>68.41\pm 1.0</u>	<u>48.8\pm 1.1</u>	72.6 \pm 0.4
AdaSpot ^b	55.38\pm 0.3	69.37\pm 0.2	69.94\pm 0.2	51.66\pm 0.6	<u>73.66\pm 0.4</u>

for nearby events. For SN-BAS, the most pronounced effect is observed for the cross event, which depends not only on the player interacting with the ball but also on the broader context of where the ball is headed, which is not covered by the RoI.

G.3. F3Set additional evaluation

Tab. 9 presents a further evaluation on the F3Set dataset. Both AdaSpot variants outperform F³ED across all mAP metrics and the F1 score, with substantial gains. However, on the Edit score, AdaSpot performs slightly lower, highlighting the contribution of the additional context refinement module introduced in F³ED. Overall, these results demonstrate that AdaSpot achieves strong performance even on the more fine-grained event classes featured in F3Set.

G.4. Qualitative results

Saliency maps and selected RoIs. To complement the visualizations in Sec. 4.4, we provide in the Supplementary Material two example clips per dataset corresponding to the best-performing AdaSpot version, showing both the saliency maps and the selected RoIs. In **Tennis** (*Video_SaliencyRoIs_Tennis_1.mp4* and *Video_SaliencyRoIs_Tennis_2.mp4*), as previously discussed, events revolve around the ball. We observe that, in most frames, the areas of highest saliency –and consequently the selected RoIs– align closely with the ball’s position. In a few cases, such as when the ball is in the air with multiple frames before or after an action, saliency occasionally shifts toward the players. However, as the clip progresses and approaches an action, the saliency consistently returns to the ball. Additionally, the generated RoIs move smoothly across frames, which facilitates effective spatio-temporal modeling within the high-resolution extractor. In **FineDiving** (*Video_SaliencyRoIs_FineDiving_1.mp4* and *Video_SaliencyRoIs_FineDiving_2.mp4*), where events center on a single athlete performing a dive, the RoIs consistently capture the athlete in nearly all frames while moving smoothly throughout the clip, demonstrating robust and reliable RoI localization. In **FineGym** (*Video_SaliencyRoIs_FineGym_1.mp4* and *Video_SaliencyRoIs_FineGym_2.mp4*), events again focus

on a single athlete, and the saliency maps reliably highlight regions including the athlete. However, the camera views in this dataset are more varied, with some closer shots resulting in RoIs that cover only part of the athlete. In these cases, the selected regions tend to focus on the most relevant parts for the event (*e.g.*, the hands contacting the vault during a vault, or the feet and floor when landing from a jump). We hypothesize that such closer views may explain why AdaSpot achieves slightly more modest results on this dataset, as the downsampled full-view frames already contain much of the necessary fine-grained detail. In **F3Set** (*Video_SaliencyRoIs_F3Set_1.mp4* and *Video_SaliencyRoIs_F3Set_2.mp4*), which resembles the Tennis dataset, we observe similar patterns: the highest saliency and selected RoIs closely align with the ball’s position. Finally, in **SN-BAS** (*Video_SaliencyRoIs_SNBAS_1.mp4* and *Video_SaliencyRoIs_SNBAS_2.mp4*), events are again ball-centric. In both clips, the saliency maps and selected RoIs consistently follow the ball, producing semantically meaningful regions. Only in frames without nearby events, saliency spreads more evenly across the scene, occasionally resulting in the ball falling outside the RoI.

AdaSpot predictions. We additionally provide, in the Supplementary Material, one example clip per dataset showing AdaSpot’s predictions, together with the temporal distance errors relative to the corresponding ground-truth annotations. In **Tennis** (*Video_Predictins_Tennis.mp4*), the strong performance reported in Tab. 1 of the main paper is clearly reflected visually: all actions are detected with high temporal precision. In **FineDiving** (*Video_Predictins_FineDiving.mp4*), all actions are still correctly identified, although some exhibit larger temporal localization errors. In **FineGym** (*Video_Predictins_FineGym.mp4*), we observe occasionally multiple predictions around a single ground-truth event, which stem from the ambiguity in localizing certain event types. Finally, in **SN-BAS** (*Video_Predictins_SNBAS.mp4*), predictions generally follow the ground truth well, achieving good precision under the more relaxed ES evaluation setting, with only one missed action near the end of the clip.

References

- [1] Bilal Alsallakh, Narine Kokhlikyan, Vivek Miglani, Jun Yuan, and Orion Reblitz-Richardson. Mind the pad—cnns can develop blind spots. *arXiv preprint arXiv:2010.02178*, 2020. 4
- [2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3
- [3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms—improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 7
- [4] A. Cioppa, S. Giancola, V. Somers, V. Joos, F. Magera, J. Held, S. A. Ghasemzadeh, X. Zhou, K. Seweryn, M. Kowalczyk, Z. Mróz, S. Lukasik, M. Haloń, H. Mkhallati, A. Deliège, C. Hinojosa, K. Sanchez, A. M. Mansourian, P. Miralles, O. Barnich, C. De Vleeschouwer, A. Alahi, B. Ghanem, M. Van Droogenbroeck, A. Gorski, A. Clapés, A. Boiarov, A. Afanasiev, A. Xarles, A. Scott, B. Lim, C. Yeung, C. Gonzalez, D. Rüfenacht, E. Pacilio, F. Deuser, F. S. Altawijri, F. Cachón, H. Kim, H. Wang, H. Choe, H. J. Kim, I.-M. Kim, J.-M. Kang, J. Tursunboev, J. Yang, J. Hong, J. Lee, J. Zhang, J. Lee, K. Zhang, K. Habel, L. Jiao, L. Li, M. Gutiérrez-Pérez, M. Ortega, M. Li, M. Lopatto, N. Kasatkin, N. Nemtsev, N. Oswald, O. Udin, P. Kononov, P. Geng, S. G. Alotaibi, S. Kim, S. Ulasen, S. Escalera, S. Zhang, S. Yang, S. Moon, T. B. Moeslund, V. Shandyba, V. Golovkin, W. Dai, W. Chung, X. Liu, Y. Zhu, Y. Kim, Y. Li, Y. Yang, Y. Xiao, Z. Cheng, and Z. Li. Soccernet 2024 challenges results. *arXiv preprint arXiv:2409.10587*, 2024. 2
- [5] Anthony Cioppa, Silvio Giancola, Vladimir Somers, Floriane Magera, Xin Zhou, Hassan Mkhallati, Adrien Deliège, Jan Held, Carlos Hinojosa, Amir M Mansourian, et al. Soccernet 2023 challenges results. *Sports Engineering*, 27(2): 24, 2024. 2
- [6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3
- [7] Mohamad Dalal, Artur Xarles, Anthony Cioppa, Silvio Giancola, Marc Van Droogenbroeck, Bernard Ghanem, Albert Clapés, Sergio Escalera, and Thomas B Moeslund. Action anticipation from soccernet football video broadcasts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6080–6091, 2025. 2
- [8] Adrien Deliège, Anthony Cioppa, Silvio Giancola, Meisam J Seikavandi, Jacob V Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4508–4519, 2021. 1
- [9] James Hong, Haotian Zhang, Michaël Gharbi, Matthew Fisher, and Kayvon Fatahalian. Spotting temporally precise, fine-grained events in video. In *European Conference on Computer Vision*, pages 33–51. Springer, 2022. 1, 2, 6
- [10] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–814, 2015. 3
- [11] Huabin Liu, Weixian Lv, John See, and Weiyao Lin. Task-adaptive spatial-temporal video sampler for few-shot action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6230–6240, 2022. 3, 4
- [12] Zhaoyu Liu, Kan Jiang, Murong Ma, Zhe Hou, Yun Lin, and Jin Song Dong. F3 set: Towards analyzing fast, frequent, and fine-grained events from videos. *arXiv preprint arXiv:2504.08222*, 2025. 1, 8

- [13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. [2](#)
- [14] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, pages 850–855. IEEE, 2006. [7](#)
- [15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. [2](#)
- [16] Sanchayan Santra, Vishal Chudasama, Pankaj Wasnik, and Vineeth N Balasubramanian. Precise event spotting in sports videos: Solving long-range dependency and class imbalance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3163–3172, 2025. [2](#), [6](#), [7](#)
- [17] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625, 2020. [1](#)
- [18] SoccerNet. Soccernet ball action spotting. <https://www.soccer-net.org/tasks/ball-action-spotting>, 2023. Online; accessed 2025-13-10. [1](#)
- [19] Kim Hoang Tran, Phuc Vuong Do, Ngoc Quoc Ly, and Ngan Le. Unifying global and local scene entities modelling for precise action spotting. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024. [2](#), [6](#)
- [20] Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. AdaFocus v2: End-to-end training of spatial dynamic networks for video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20030–20040. IEEE, 2022. [3](#)
- [21] Yulin Wang, Haoji Zhang, Yang Yue, Shiji Song, Chao Deng, Junlan Feng, and Gao Huang. Uni-adafocus: spatial-temporal dynamic computation for video recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. [3](#), [4](#), [6](#)
- [22] Artur Xarles, Sergio Escalera, Thomas B Moeslund, and Albert Clapés. T-deed: Temporal-discriminability enhancer encoder-decoder for precise event spotting in sports videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3410–3419, 2024. [2](#), [7](#)
- [23] Artur Xarles, Sergio Escalera, Thomas B Moeslund, and Albert Clapés. T-deed revisited: Broader evaluations and insights in precise event spotting. 2024. [6](#)
- [24] Jinglin Xu, Yongming Rao, Xumin Yu, Guangyi Chen, Jie Zhou, and Jiwen Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2949–2958, 2022. [1](#)
- [25] Haotian Zhang, Cristobal Sciutto, Maneesh Agrawala, and Kayvon Fatahalian. Vid2player: Controllable video sprites

that behave and appear like professional tennis players. *ACM Transactions on Graphics (TOG)*, 40(3):1–16, 2021. [1](#)