T-DEED Revisited: Broader Evaluations and Insights in Precise Event Spotting

Artur Xarles[1,2*], Sergio Escalera[1,2,3], Thomas B. Moeslund[3], Albert Clapés[1,2]

[1]Universitat de Barcelona, Barcelona, 08034, Spain.
[2]Computer Vision Center, Cerdanyola del Vallès, 08193, Spain.
[3]Aalborg University, Aalborg, 9220, Denmark.

*Corresponding author(s). E-mail(s): arturxe@gmail.com;
Contributing authors: sescalera@ub.edu; tbm@create.aau.dk;
aclapes@ub.edu;

Abstract

In this paper, we introduce **T-DEED**, a Temporal-Discriminability Enhancer Encoder-Decoder for Precise Event Spotting (PES) in sports videos. T-DEED addresses multiple challenges in the task, including the need for discriminability among frame representations, high output temporal resolution to maintain prediction precision, and the necessity to capture information at different temporal scales to handle events with varying dynamics. It tackles these challenges through its specifically designed architecture, featuring an encoder-decoder for leveraging multiple temporal scales and achieving high output temporal resolution, along with temporal modules designed to increase token discriminability. Leveraging these characteristics, T-DEED achieves state-of-the-art (SOTA) performance on four PES datasets: FigureSkating, FineDiving, FineGym and Tennis. Additionally, it excels in the broader Action Spotting task, achieving top results on the SoccerNet Action Spotting dataset using raw input frames – without relying on pre-extracted features – and securing 1st place in the 2024 SoccerNet Ball Action Spotting challenge. The code is available at https://github.com/arturxe2/T-DEED.

Keywords: Video Understanding, Precise Event Spotting, Action Spotting, Discriminability

1 Introduction

Recent advancements in deep learning and computational power have driven remarkable progress in video understanding. These advancements have enabled researchers to move beyond simpler tasks like action recognition in trimmed videos [1] to more complex challenges such as accurate localization of actions within untrimmed videos. These tasks include Temporal Action Localization (TAL), which represents actions as temporal intervals, and Action Spotting (AS), which uses single keyframes. While TAL [2] has historically received more attention, AS has recently gained interest, particularly in domains such as sports, as fast-paced actions – common in sports videos – are often better represented by single temporal positions rather than attempting to determine both the beginning and end of an action. This approach reduces the annotation burden by requiring only a single time mark per action. Moreover, Hong et al. [3] expanded AS into Precise Event Spotting (PES), using tighter evaluation tolerances and broadening the concept of actions to more general events (i.e., not requiring to be triggered by an agent). In the context of sports, since most events are agent-triggered, we will refer to events and actions interchangeably.

This paper serves as an extension of our previous work [4] and specifically focuses on the task of Precise Event/Action Spotting in sports, as illustrated in Figure 1. We conduct our evaluation on four sports datasets: FigureSkating [5] – which includes two different splits, FS-Comp and FS-Split –, FineDiving [6], FineGym [7], and Tennis [8]. Following Hong et al. [3], we use tight tolerances to accommodate the fast-paced nature of sporting events, where even a small temporal deviation of 1-2 frames can lead to missed events. Three main challenges faced by methods addressing this task include: (1) the need for discriminative per-frame representations to differentiate between frames with high spatial similarity, particularly when they correspond to different event labels, (2) the necessity of high output temporal resolution to avoid losing prediction precision, and (3) the variability in the amount of temporal context required for different events, influenced by dataset characteristics and event dynamics.

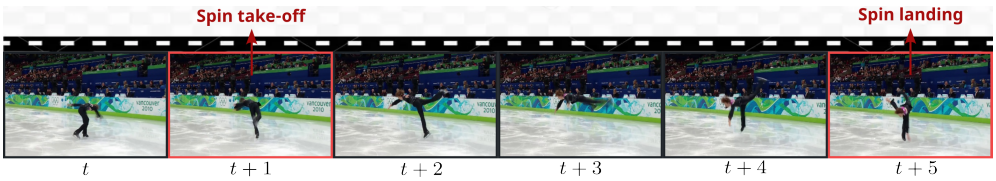


Fig. 1: Illustration of the Precise Event Spotting task on the FigureSkating dataset. Red-marked frames contain events that require precise localization and correct classification among possible classes.

To address these challenges, we introduce T-DEED, a Temporal Discriminability Enhancer Encoder-Decoder for PES in sports videos. By incorporating skip connections within its encoder-decoder architecture, T-DEED operates across various temporal scales, capturing actions that require diverse temporal contexts while restoring the original temporal resolution, thereby tackling challenges (2) and (3).

Additionally, it integrates Scalable-Granularity Perception (SGP) [9] based layers to increase discriminability among features within the same temporal sequence, addressing challenge (1). To summarize, our main contributions are:

1. We integrate residual connections into the SGP layer, enabling the fusion of features from multiple temporal scales within the skip connections of our encoder-decoder architecture. This results in our proposed SGP-Mixer layer, which addresses challenge (3).
2. We introduce the SGP-Mixer module within the SGP-Mixer layer, which adapts the SGP module to aggregate information from different temporal scales. This module shares the core principles of SGP to promote token discriminability while modeling temporal information, therefore tackling challenge (1).
3. We conduct extensive ablation studies on T-DEED components, highlighting the advantages of the encoder-decoder architecture combined with SGP-based layers to enhance token discriminability. Furthermore, T-DEED achieves state-of-the-art (SOTA) performance across four PES datasets, with improvements in mean Average Precision (mAP) at a 1-frame tolerance: +1.15 on FineDiving, +3.07 on FS-Comp, +4.83 on FS-Perf, +0.55 on Tennis, and +4.84 on FineGym.
4. We extend the evaluation of T-DEED to the broader Action Spotting task, achieving the best results among methods that do not rely on pre-extracted features in the SoccerNet Action Spotting (SN-AS) dataset. We also secure 1st place in the 2024 SoccerNet Ball Action Spotting (SN-BAS) challenge, improving the previous baseline by +17.24 points and outperforming the second-best method by +2.04 points.

In the following sections, we detail related work (Section 2), introduce our proposed T-DEED method (Section 3), present the results and ablations (Section 4), and conclude the paper (Section 5).

2 Related work

Over the past decade, deep learning has driven the field of video understanding through a remarkable evolution. Initially focused on simple tasks like classifying short-trimmed videos [10–12], the field has transitioned to more complex challenges, including Temporal Action Localization (TAL) and Action Spotting (AS). Both TAL and AS focus on temporally locating specific actions within untrimmed videos. TAL specifies temporal intervals for annotations, while AS represents actions with single keyframes, making TAL suitable for prolonged actions and AS more appropriate for fast-occurring actions, with the added benefit of reduced annotation costs. Furthermore, Hong et al. [3] extended the AS task to Precise Event Spotting (PES), introducing a key difference in the required precision of predictions, limited to only a few frames, and distinguishing between actions and events. In the sports domain, AS and PES are particularly popular as they adapt better to its fast nature, leading to the development of many approaches [3, 13–17] across different sports, including football [18, 19], figure skating [5], diving [6], gymnastics [7], and tennis [8].

Given the inherent similarities between TAL and AS, the methodologies developed for these tasks frequently share common components. However, TAL methods have attracted more attention due to the earlier introduction of the task and a more extensive set of benchmarking datasets [20–24], placing them a step ahead of AS methods. In contrast, AS methods have been tailored for specific datasets and competitive challenges, exemplified by their development in challenges like SoccerNet Action Spotting and SoccerNet Ball Action Spotting [15, 16]. Notably, E2E-Spot [3] stands out as the only action or event spotting method evaluated across multiple datasets.

Temporal Action Localization. In TAL methods, a common classification divides them into two groups: two-stage methods [25–29] and one-stage methods [9, 30–33]. Two-stage methods generate class-agnostic proposals that are later classified into action labels or background, while one-stage models directly localize and classify actions in a single step, offering simplicity and achieving SOTA performance in many TAL and AS scenarios.

Among one-stage methods, early approaches [34, 35] utilized anchor windows to generate action predictions. Later, Yang et al. [36] introduced an anchor-free approach, relying on temporal points instead of anchor windows, highlighting the advantages of both techniques. Building on this approach, current SOTA methods such as ActionFormer [32] and TriDet [9] have exhibited remarkable performance across various datasets. They leverage a feature pyramid network to process features at different temporal scales, a critical aspect for identifying actions that require distinct temporal contexts. The main difference lies in their prediction head and the layers used for feature processing. ActionFormer employs transformer layers, later revealed to suffer from the rank loss problem [37], negatively impacting token discriminability. To alleviate this problem, TriDet proposes a more efficient convolutional-based Scalable-Granularity Perception (SGP) layer, specially designed to increase token discriminability within the same temporal sequence, contributing to an improved overall performance. T-DEED, inspired by TAL trends, focuses on enhancing token discriminability while leveraging multiple temporal scales, with modifications tailored to meet the precision requirements of PES.

Action Spotting. In AS, techniques similar to those in TAL have demonstrated SOTA performance on the SoccerNet challenges [15, 16]. Many methods [13, 14, 17] classify temporal points as either background or actions and refine them through temporal regression using either convolutional [13] or Transformer-based [14, 17] approaches. In contrast, Hong et al. [3] propose a simple end-to-end solution that employs a convolutional backbone with Gate-Shift Modules (GSM) [38] for extracting per-frame features with short-term temporal information, followed by a Gated Recurrent Unit (GRU) [39] layer for long-term temporal information. This approach proves effective in their proposed task of PES across four different datasets [5–8], and is also adaptable to the coarser AS task in SoccerNet.

While many TAL and AS methods [9, 13, 14, 32] rely on pre-extracted features due to their efficiency in training, end-to-end approaches have demonstrated that they can be beneficial in learning more meaningful features in some cases. This is exemplified by Hong et al. [3], particularly in scenarios where precise predictions are essential, such as in the case of PES. Exploiting these advantages, T-DEED also adopts an end-to-end approach.

3 Method

Problem definition. Precise Event Spotting (PES) involves the identification and localization of events within an untrimmed video X, as illustrated in Figure 1. Given the video input, the objective is to recognize and locate all the events occurring in the video, represented as $E = \{e_1, \dots, e_N\}$. The number of events, denoted as N, may vary across different videos. Each event instance e_i comprises an action class $c_i \in \{1, \dots, C\}$ (where C is the total number of distinct event classes) and its corresponding temporal position t_i (i.e. the exact frame where it occurs), forming a pair $e_i = (c_i, t_i)$.

Method overview. Our model, Temporal-Discriminability Enhancer Encoder Decoder (T-DEED), is designed to increase token discriminability for Precise Event Spotting (PES) while leveraging multiple temporal scales. As illustrated in Figure 2, T-DEED comprises three main blocks: a feature extractor, a temporally discriminant encoder-decoder, and a prediction head. We process videos through fixed-length clips, each containing L densely sampled frames. The feature extractor, composed of a 2D backbone with Gate-Shift-Fuse (GSF) modules [40], handles the input frames, generating per-frame representations of dimension d, hereby referred to as *tokens*. These tokens undergo further refinement within the temporally discriminant encoder-decoder. This module employs SGP layers which – as shown by Shi et al. [9] – diminish token similarity, thereby boosting discriminability across tokens within the same sequence. The encoder-decoder architecture allows the processing of features across diverse temporal scales, helpful for detecting events requiring different amounts of temporal context. We also integrate skip connections to preserve the fine-grained information from the initial layers in subsequent stages of the model. To effectively merge information proceeding from varying temporal scales in the skip connections, we introduce the SGP-Mixer layer, detailed in Section 3.2. This layer employs the same principles as the SGP layer to promote token discriminability while gathering information from varying range temporal windows. Finally, the output tokens are directed to the prediction head, resembling those commonly used in Action Spotting (AS) literature [13, 14, 17]. It encompasses a classification component to identify whether an event occurs at the given temporal position or in close proximity (within a radius of r_E frames). Additionally, for the positive classifications, a displacement component pinpoints the exact temporal position of ground truth events.

Further details of the proposed method are discussed in the following sections.

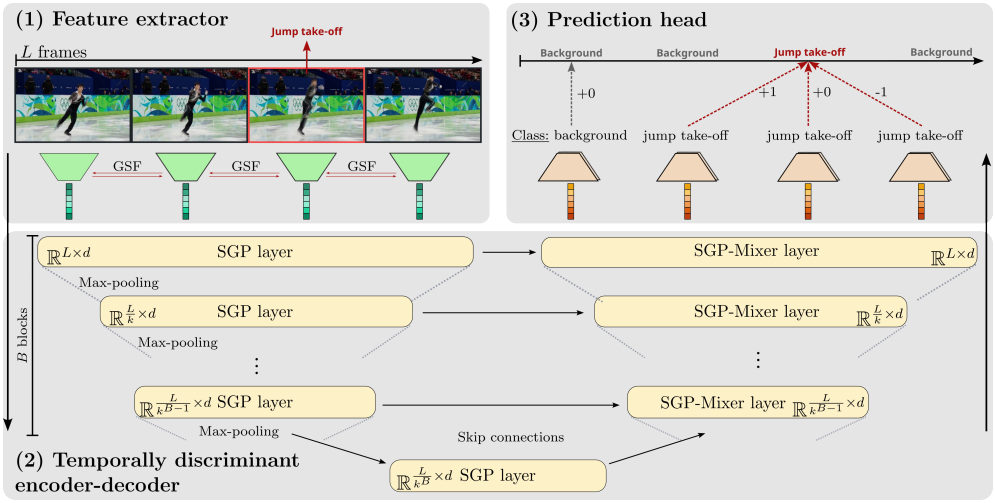


Fig. 2: Illustration of **T-DEED** architecture comprising three key components: (1) **Feature extractor** to produce per-frame representations, (2) **Temporally discriminant encoder-decoder** to capture local and global temporal information while promoting token discriminability, and (3) **Prediction head** to generate per-frame classifications and displacements for refinement.

3.1 Feature extractor

The feature extractor processes the input frame sequence, $\mathbb{R}^{L\times H\times W\times 3}$ with $H \times W$ denoting the spatial resolution, and produces per-frame feature representations, $\mathbb{R}^{L\times d}$. Following Hong et al. [3], we employ a compact 2D backbone to ensure efficiency and to accommodate longer video sequences. For comparability with previous PES methods, we choose RegNetY [41] as our feature extractor, known for its efficiency. To incorporate local temporal information during the extractor, we utilize GSF [40], which overperforms GSM [38] in action recognition tasks. GSF modules are specifically applied to the latter half of the RegNetY backbone, allowing for spatial-only modeling initially, before integrating local temporal context.

3.2 Temporally discriminant encoder-decoder

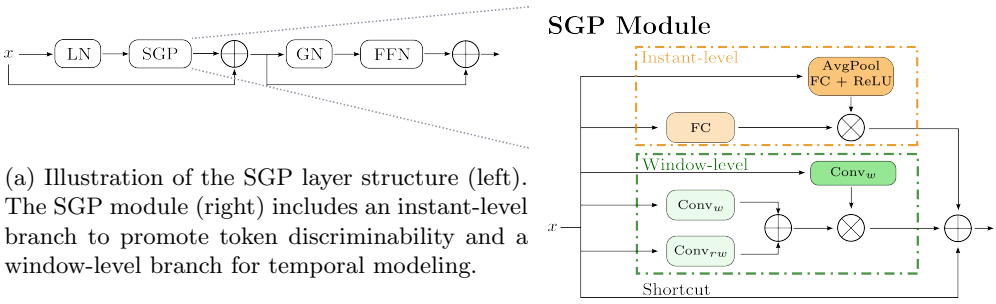
The temporally discriminant encoder-decoder processes the tokens produced by the feature extractor module, which contain mainly spatial information, with the objective of enriching them with essential temporal information – both local and global – to enable precise event predictions. To accomplish this, it incorporates three key components: (1) an encoder-decoder architecture for exploiting information across diverse temporal scales, (2) SGP layers to promote discriminability among tokens within the same sequence, and (3) our novel SGP-Mixer layer, designed to effectively fuse features from varying temporal scales while maintaining the distinctiveness of its output tokens.

Encoder-decoder architecture. Various works [9, 13, 32] have shown the benefits of processing features at different temporal scales in tasks related to action recognition. This is because certain actions inherently require longer temporal context for recognition and localization, while others can be identified with just a few frames. SOTA models in TAL achieve this by employing a feature pyramid, where the temporal dimension is downscaled by a factor of k in the final layers through max-pooling, and predictions are made for each output feature at various temporal scales. However, in PES, where precision is crucial, this is detrimental. As shown in Section 4.6, the further down we go in the feature pyramid, the more deminishes the precision of the output predictions, impacting model performance. To address this issue while still leveraging different temporal scales, we propose an encoder-decoder architecture. After temporal downscaling during the encoder, akin to the feature pyramid, we restore the original temporal resolution through the decoder, thereby regaining frame-level granularity for the representations. We do this by incorporating skip connections from the encoder to the decoder's upsampling.

Specifically, the encoder begins by complementing the tokens with a learnable encoding that specifies its temporal position. Furthermore, it comprises B encoder blocks, each consisting of an SGP layer to promote discriminability while capturing temporal context, and a max-pooling operation to reduce the temporal dimension by a factor of k. An additional SGP layer is applied in the neck of the encoder-decoder, before passing the features to the decoder. In the decoder, the original temporal resolution is restored through B decoder blocks. Each of these blocks incorporates an SGP-Mixer layer, which extends the SGP layer to increase the temporal dimension by a factor of k while integrating information coming from its corresponding skip connection.

SGP layer. The SGP layer, as introduced by Shi et al. [9], addresses the rank-loss problem [37] commonly encountered in Transformers [42], thus improving token discriminability within sequences. As shown on the left of Figure 3, it replaces the self-attention module in Transformer layers with a SGP module (depicted on the right). This is a convolutional-based module that comprises two primary branches: an instant-level branch and a window-level branch. The instant-level branch aims to promote token discriminability by comparing each token to the clip-level average token, adjusting the token's distinctiveness whenever beneficial for the network. The window-level branch captures temporal information from multiple receptive fields. Furthermore, the SGP layer replaces one of the layer normalization modules with group normalization.

Given its advantageous characteristics, which contributed to Tridet's SOTA results in TAL, we believe it can be even more beneficial in PES, where precise frame-level predictions are crucial and can benefit from improved discriminability between concurrent tokens, as discussed in Section 4.6.



(a) Illustration of the SGP layer structure (left). The SGP module (right) includes an instant-level branch to promote token discriminability and a window-level branch for temporal modeling.

Fig. 3: SGP Layer.

More formally, the SGP module can be defined as:

$$f_{SGP}(x) = IB(x) + WB(x) + x, \tag{1}$$

where $IB(\cdot)$ represents the instant-level branch operations, and $WB(\cdot)$ represents the window-level branch operations, defined as follows:

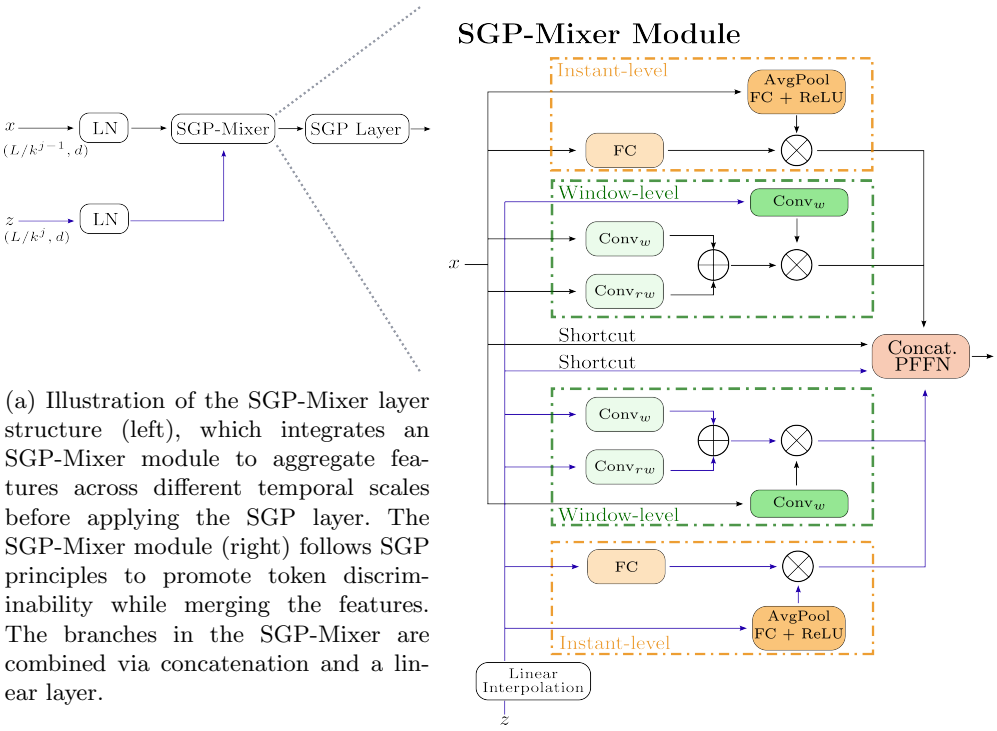
$$IB(a) = \phi(a) \otimes FC(a), \quad \text{with} \quad \phi(a) = ReLU(FC(AvgPool(a))) \tag{2}$$

$$WB(a) = \psi(a) \otimes (Conv_w(a) + Conv_{rw}(a)), \quad \text{with} \quad \psi(a) = Conv_w(a) \tag{3}$$

Here, $Conv_k$ represents a convolutional layer with a kernel size of k, FC is a fully-connected layer, and $\otimes$ indicates element-wise tensor multiplication.

SGP-Mixer layer. As shown on the left in Figure 4, the SGP-Mixer layer extends the original SGP layer to accommodate two inputs with distinct temporal scales. This adaptation becomes necessary in the decoder due to the reception of an input feature $z \in \mathbb{R}^{L/k^j \times d}$ from the preceding decoder layer, and encoder features $x \in \mathbb{R}^{L/k^{(j-1)} \times d}$ via the skip connection, where $j \in \{1, \ldots, B\}$ denotes the depth within the decoder blocks. Both features undergo layer normalization before entering the SGP-Mixer module, where a combination of both features is generated. Finally, the output features are further processed through identical components within an SGP layer.

The fusion of both features is done in the SGP-Mixer module as detailed on the right of Figure 4. First, the features from the previous layer are upsampled to match the temporal dimensions ($\mathbb{R}^{L/k^j} \rightarrow \mathbb{R}^{L/k^{(j-1)}}$) using linear interpolation. With both features now sharing the same temporal resolution, the SGP-Mixer module – building on the principles of the SGP layer – utilizes two instant-level branches, one for each input feature, to promote token discriminability. This is particularly important for the features coming from the previous layer, since upsampling may produce similar tokens in adjacent temporal positions. In addition, it includes two window-level branches to aggregate information from both features while capturing different temporal contexts. In each branch, one feature evolves to mix information across different temporal receptive fields, which is later gated by the other feature. Shortcuts are also



(a) Illustration of the SGP-Mixer layer structure (left), which integrates an SGP-Mixer module to aggregate features across different temporal scales before applying the SGP layer. The SGP-Mixer module (right) follows SGP principles to promote token discriminability while merging the features. The branches in the SGP-Mixer are combined via concatenation and a linear layer.

Fig. 4: SGP-Mixer Layer.

introduced, but unlike SGP, feature aggregation from different branches involves concatenation and linear projection, which we find more effective than simple addition, as will be shown later in Section 4.6.

More formally, the SGP-Mixer can be defined as:

$$f_{SGP-Mixer}(x, z) = PFFN(\text{Concat}[IB(x), IB(LI(z)), WB(x, LI(z)), \\ WB(LI(z), x), LI(z), x]) \quad (4)$$

where $LI(\cdot)$ is the linear interpolation operation, $Concat[\cdot]$ denotes concatenation operation along the features' dimension, and $PFFN$ refers to a point-wise feed-forward network with two fully-connected layers. $IB(\cdot)$ and $WB(\cdot)$ are the instant-level and window-level operations as in the SGP module, with the window-level branch (WB) now modified to accept two inputs, defined as:

$$WB(a, b) = \psi(a) \otimes (Conv_w(b) + Conv_{rw}(b)), \quad (5)$$

3.3 Prediction head

Following common AS approaches [13, 14, 17], we include a prediction head consisting of a classification head and a displacement head. The classification head uses a linear

layer and a softmax activation to project the output of the temporally discriminant encoder-decoder, $\mathbb{R}^{L\times d}$, onto $\hat{y}^c \in \mathbb{R}^{L\times(C+1)}$, representing the probability of each temporal position containing each of the events or a background class. Similarly, the displacement head uses a linear layer to project the same output to $\hat{y}^d \in \mathbb{R}^{L\times 1}$, representing the displacement toward the ground truth event if an actual event is present at the corresponding temporal position.

3.4 Training details

The model is trained using a combination of a classification loss ($\mathcal{L}_c$) and a displacement loss ($\mathcal{L}_d$). For classification, per-frame cross-entropy loss is employed weighting the positive classes by a factor of w for all events. Mean squared error is used for displacement. The final loss ($\mathcal{L}$) for a given clip is the sum of both losses:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_d = \frac{1}{L}\sum_{l=1}^{L} CE_w(y_l^c, \hat{y}_l^c) + MSE(y_l^d, \hat{y}_l^d), \tag{6}$$

where y_l^c represents the one-hot encoding of the event in frame l, $\hat{y}_l^c$ denotes the classification probabilities at that frame, y_l^d indicates the actual displacement to a ground-truth event (if an event is within the detection radius r_E), and $\hat{y}_l^d$ represents the predicted displacement.

During training, unless otherwise specified, clips are randomly sampled from the training split, and standard data augmentation techniques are applied across all datasets, including random cropping, horizontal flipping, Gaussian blur, color jitter, and mixup [43].

3.5 Inference

At inference time, the data augmentation techniques are disabled and we use clips with 75% of overlapping. Moreover, to reduce the number of candidate events, Soft Non-Maximum Suppression (S-NMS) [44] is applied.

4 Results

In this section, we detail the evaluation setup for our proposed method and present experimental results demonstrating its superiority over current SOTA methods in PES and AS on the application domain of sports data. Additionally, we conduct ablations on key components of our proposal.

4.1 Datasets

We conduct experiments on six different sports datasets, four dedicated to PES and two to the broader AS task:

- The **FigureSkating** [5] dataset contains 11 videos featuring 371 short program performances from the Winter Olympics and World Championships, recorded at 25 frames per second (fps) and annotated with four different event classes. Following

Hong et al. [3], we use two different splits: the *Competition Split* (FS-Comp) and the *Performance Split* (FS-Perf).

- The **FineDiving [6]** dataset consists of 3000 diving clips at 25 fps, annotated with four different event classes representing transitions between the different phases of a dive.
- The **Tennis** [8] dataset, initially introduced in Vid2Player [8] and later extended by Hong et al. [3], includes 3345 video clips from 28 tennis matches, each showing a single point and annotated with six event classes. The frame rates range between 25 and 30 fps.
- The **FineGym [7]** dataset consists of 5374 untrimmed gymnastics videos, annotated with 32 different events. While the original videos had frame rates between 25 and 60 fps, we follow Hong et al. [3] and resample some videos to standardize the frame rates between 25 and 30 fps. As in E2E-Spot, we report performance on both the full set of classes (FG-Full) and a subset with only start-of-event classes (FG-Start).
- The **SoccerNet Action Spotting (SN-AS) [19]** dataset includes 550 football game broadcasts at 25 fps, with sparse annotations covering 17 different actions.
- The **SoccerNet Ball Action Spotting (SN-BAS) [19]** dataset contains 9 football matches recorded with a single camera and densely annotated with 12 ball-related actions. The videos are recorded at 25 fps.

Tables 8, 9, 10, 11, 12, and 13 show the different classes for each dataset and the total number of observations across the train, validation, and test splits. As shown, for the four PES datasets, most actions have a decent number of observations, except for the last two events in FineGym, which have only 49 observations. In contrast, SN-AS and SN-BAS contain several actions with very few examples, making them more challenging for the model to learn.

4.2 Action Spotting (AS) adaptation from Precise Event Spotting (PES)

Adapting T-DEED from PES to the broader AS tasks in the SoccerNet datasets is relatively straightforward due to the similarity between the tasks. The main adjustment involves tuning the model hyperparameters to handle longer clips at a lower frame rate. However, for the SN-BAS dataset, which suffers from limited data for some classes, additional adaptations are necessary to enhance performance.

SN-BAS adaptations. As detailed in Section 4.1, the SN-BAS dataset is relatively small, with only 9 games and some actions occurring infrequently, complicating the training process due to the limited number of examples. To mitigate this issue, we train T-DEED using both the SN-BAS dataset and the larger, related SN-AS dataset, which contains 550 games and shares some overlapping actions. We achieve this by incorporating two prediction heads into our architecture – one for each dataset – allowing T-DEED to train in a multi-task manner and improve performance on SN-BAS through the complementary data provided by SN-AS. During training, half of the clips are sampled from SN-BAS and the other half from SN-AS. The final loss

is computed by summing the individual losses from each dataset after applying their respective prediction heads.

4.3 Evaluation

Following standard practices, we train the models on the training splits, use the validation splits for early stopping, and evaluate them on the test splits for FineDiving, FigureSkating, Tennis, FineGym, and SN-AS. However, to participate in the SN-BAS challenge, we utilize all available data (train, validation, and test splits) to train the model and evaluate it on the challenge split with unknown labels. Model performance is assessed using the mAP metric, which calculates the mean of Average Precisions across different events. For FineDiving, FigureSkating, Tennis, and FineGym, we evaluate two versions of the metric: a tight mAP metric with a tolerance of $\delta = 1$ frame and a loose version with $\delta = 2$ frames. For SN-AS, the mAP is averaged over tolerances ranging from 0.5 to 2.5 seconds[1], with 0.5-second increments for the tight metric, and from 2.5 to 30 seconds in 2.5-second increments for the loose metric. For SN-BAS, a tolerance of 0.5 seconds is applied.

4.4 Implementation details

We train T-DEED using clips of $L = 100$ frames sampled at the original frame rate for FineDiving, FigureSkating, Tennis, and FineGym, at 2 fps for SN-AS, and at 12.5 fps for SN-BAS, with a batch size of 8 clips. Frame resolution is set to 398×224 for FigureSkating, Tennis, and SN-AS; 224×224 for FineDiving and FineGym; and 796×448 for SN-BAS, with random cropping to 224×224 for FigureSkating and Tennis. Each epoch consists of 5000 clips randomly sampled from the training videos, except for SN-BAS, where it is doubled since we train on both SN-BAS and SN-AS data simultaneously. The models are trained for 50 epochs on FineDiving, FigureSkating and Tennis datasets, 100 epochs for the larger FineGym and SN-AS datasets, and 35 epochs for SN-BAS. We use the AdamW optimizer [45] with a base learning rate of 8e-4, incorporating 3 linear warmup epochs followed by cosine decay. Positive classes in the cross-entropy loss are weighted by a factor $w = 5$ in order to compensate for the presence of the background (no action) class. We evaluate two versions of our feature extractor, RegNetY-200MF and RegNetY-800MF, with hidden dimensions set to $d = 368$ and $d = 768$ respectively, and a max-pooling stride (downscaling factor) of $k = 2$. Additional dataset-specific hyperparameters can be found in the supplementary material.

4.5 Comparison to SOTA

Precise Event Spotting task. In Table 1, we compare our proposed T-DEED in two variations – utilizing smaller and larger feature extractors – against previous SOTA models across six configurations: FS-Comp, FS-Perf, FineDiving, Tennis, FG-Full,

[1]The metric used aligns with the SoccerNet evaluation protocols, though the reported tolerance values may differ. In our case, δ is defined such that an action is considered correct if it falls within the interval $[t_{gt} - \delta, t_{gt} + \delta]$. In other approaches, the tolerance is typically interpreted as the interval $[t_{gt} - \frac{\delta}{2}, t_{gt} + \frac{\delta}{2}]$, where t_{gt} represents the temporal position of the ground truth action.

Model	Size	OF	FS-Comp		FS-Perf		FineDiving	
			$\delta = 1$	2	1	2	1	2
E2E-Spot [3]	200MF		81.0	<u>93.5</u>	85.1	95.7	68.4	85.3
E2E-Spot [3]	800MF		84.0	-	83.6	-	64.6	-
E2E-Spot [3]	800MF	✓	83.4	**94.9**	83.3	<u>96.0</u>	66.4	84.8
T-DEED	200MF		**85.15**	91.70	<u>86.79</u>	**96.05**	<u>71.48</u>	<u>87.62</u>
T-DEED	800MF		<u>84.77</u>	92.86	**88.17**	95.87	**73.23**	**88.88**

Model	Size	OF	Tennis		FG-Full		FG-Start	
			$\delta = 1$	2	1	2	1	2
E2E-Spot [3]	200MF		96.1	97.7	47.9	65.2	61.0	78.4
E2E-Spot [3]	800MF		96.8	-	50.1	-	-	-
E2E-Spot [3]	800MF	✓	96.9	**98.1**	51.8	68.5	65.3	<u>81.6</u>
T-DEED	200MF		<u>97.03</u>	97.92	<u>55.97</u>	<u>68.55</u>	<u>68.06</u>	81.00
T-DEED	800MF		**97.45**	<u>97.97</u>	**56.64**	**69.77**	**68.68**	**82.05**

Table 1: Comparison of state-of-the-art methods across the FigureSkating (FS-Comp and FS-Perf splits), FineDiving, Tennis, and FineGym (FG-Full and FG-Start) datasets. The best results in terms of mAP are highlighted in bold, while the second-best results are underlined. For a fair comparison, we report the feature extractor sizes in MegaFlops (MF). Models shaded in gray are not comparable due to the inclusion of Optical Flow (OF) [46] in addition to RGB images.

and FG-Start. As shown, T-DEED consistently achieves the best performance on the tight metric across all configurations. Notably, the largest improvements are observed in FG-Full and FineDiving, with gains of +4.84 and +4.83 points respectively. On the Tennis dataset, the improvement is smaller at +0.55 points, which is expected given the already saturated performance levels on this dataset. For the loose metric, with a tolerance of $\delta = 2$ frames, results vary across datasets. T-DEED remains superior or comparable to previous methods in most cases, though it falls slightly behind on FS-Perf. We attribute this to the use of dilation in the E2E-Spot model for the FigureSkating dataset, which appears to offer advantages when using larger temporal tolerances.

Additionally, in Tables 8, 9, 10, and 13, we present the per-class results for all datasets. In the FigureSkating dataset, the performance across different classes is relatively consistent, with slightly better results for takeoff classes in FS-Comp, and jump-related classes showing the highest performance in FS-Perf. In the FineDiving dataset, the model struggles the most with accurately locating the class representing the transition towards a twist. The Tennis dataset, on the other hand, shows a highly saturated performance, with exceptionally high scores across all classes, particularly for serves. Finally, the FineGym dataset presents more variability, with some classes achieving good performance while others with a really low performance. As noted by Hong et al. [3], the annotations in this dataset are not always entirely precise, which complicates both model learning and evaluation.

Action Spotting in SN-AS. In Table 2, we present results on the SN-AS dataset, focusing only on methods that have evaluated their performance on the test split, excluding those that report only challenge results, for a fair comparison. We distinguish between methods using an end-to-end approach and those relying on pre-extracted features. As demonstrated in the SoccerNet 2022 [16] and 2023 [15] challenges, Baidu features [47] – which combine outputs from five backbones fine-tuned on SoccerNet along with 77 additional non-public additional games – have boosted performance in this task. However, this approach is not easily generalizable to new games or datasets, limiting its practicality, and it is not directly compared to other approaches due to the use of non-public data. In contrast, end-to-end approaches are more general and thus more practical. As shown in the table, the highest-performing methods utilize pre-extracted features, achieving an Average-mAP of up to 73.10. Among end-to-end approaches, T-DEED outperforms E2E-Spot and is only a few points behind methods that use specialized features and architectures tailored to the dataset. This highlights T-DEED as a strong initial approach for this task, without relying on pre-extracted features or dataset-specific architectures.

Model	Baidu features	Average-mAP *tight (0.5-2.5 seconds)*	Average-mAP *loose (2.5-30 seconds)*
Zhou et al. [47]	✓	47.05	73.77
E2E-Spot (800MF) [3]		61.82	74.05
Soares et al. [13]	✓	65.10	78.50
ASTRA [14]	✓	66.82	77.09
COMEDIAN[2] [17]	✓	73.10	-
T-DEED (200MF)		61.52	71.99
T-DEED (800MF)		**63.42**	**74.97**

Table 2: Comparison of SOTA methods on the SN-AS dataset test split, reporting the Average-mAP for each method. Methods shaded in gray use pre-extracted Baidu Features and are thus not directly comparable.

Additionally, in Table 3, we compare the per-class results of our end-to-end method with the only feature-based approach that reports per-class results, ASTRA. The performance gap between our method and the feature-based approach is most noticeable in classes with low frequency of occurrence or high intra-class variability, where the additional data used to fine-tune the backbones for the Baidu features can impact the results. However, in most other classes, our results are comparable to those of the feature-based approach.

Action Spotting in SN-BAS. Finally, we evaluated T-DEED on the 2024 SN-BAS challenge. Due to the dataset's small size, as detailed in Section 4.2, we trained T-DEED jointly on SN-AS and SN-BAS. Three different submissions were made, varying in sampling strategy and evaluation protocol:

[2] Although COMEDIAN uses raw frames as input, pre-extracted Baidu features are employed during pre-training, so we classify it with methods using pre-extracted features. Additionally, inconsistencies between test and challenge split results, with the test performance higher, suggest possible overfitting on this split.

Action	Model *ASTRA*	Model *T-DEED*	Action	Model *ASTRA*	Model *T-DEED*
Ball out of play	80.70	80.21$_{-0.49}$	Kick-off	68.41	62.52$_{-5.89}$
Throw-in	78.99	79.55$_{+0.56}$	Direct free-kick	73.98	65.17$_{-8.81}$
Foul	77.89	75.49$_{-2.40}$	Offside	61.31	59.38$_{-1.93}$
Indirect free-kick	56.25	58.86$_{-2.61}$	Yellow Card	65.29	67.82$_{+2.53}$
Clearance	66.00	67.32$_{+1.32}$	Goal	84.19	81.93$_{-2.26}$
Shot on target	61.96	51.11$_{-10.85}$	Penalty	86.74	79.48$_{-7.26}$
Shot off target	65.94	58.77$_{-7.17}$	Red Card	40.42	33.37$_{-7.05}$
Corner	83.96	84.02$_{+0.06}$	YC → RC	28.46	17.69$_{-10.77}$
Substitution	55.51	59.07$_{+3.56}$			

Table 3: Per-class results of ASTRA and T-DEED on the SoccerNet AS dataset, showing the Average-AP score for each action, sorted by frequency from most to least common.

- *T-DEED(c):* Uniform sampling from all possible clips, trained on the train and test splits, with the validation split used for early stopping.
- *T-DEED(b):* Uniform sampling from SN-BAS and action-specific sampling from SN-AS, trained on the train and test splits, with the validation split used for early stopping.
- *T-DEED(a):* An ensemble of two T-DEED models – one sampling from all clips and the other from action-specific clips in SN-AS – trained on all available splits (train, validation, and test). Results were aggregated by averaging per-frame probabilities after applying the predicted displacements.

Rank	Model	mAP ($\delta = 0.5s.$)
1	**T-DEED(a)**	**73.39**
2	UniBw Munich - VIS	71.35
-	T-DEED(b)	67.92
3	FS-TAHAKOM	67.09
-	T-DEED(c)	66.55
4	MobiusLabs	62.53
5	AI4Sports	62.44
6	Team sota	62.44
7	SAIVA	56.74
8	Baseline	56.15

Table 4: SN-BAS 2024 Challenge results. For each method, we report its competition rank along with the corresponding mAP score.

As shown in Table 4, all three submissions outperform the baseline and many competitors by a large margin. Notably, when comparing T-DEED(c) and T-DEED(b), we observe that sampling directly from action-specific clips in SN-AS slightly improves performance, likely benefiting tail classes by addressing the dataset's long-tail problem. However, both methods fall short compared to our main submission, T-DEED(a),

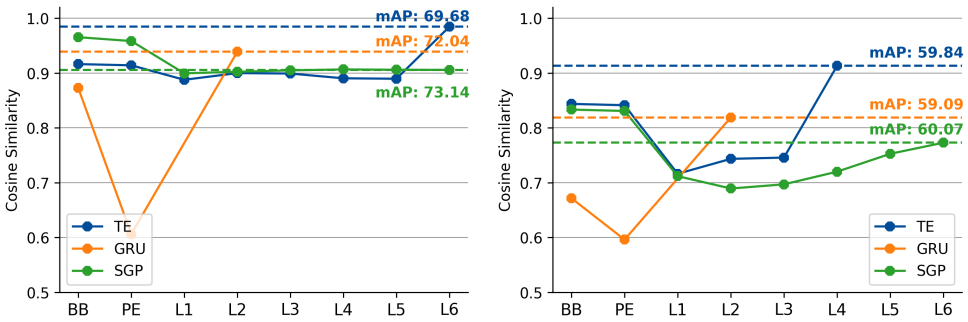
which achieved first place in the challenge with a mAP of 73.39 – outperforming the second place by +2.04 points and the baseline by +17.24 points. Additionally, T-DEED(a) shows a +5.38 point improvement over our previous submissions, likely due to training on all available data (train, validation, and test splits), which is crucial given the dataset's size. The ensemble approach in T-DEED, combining predictions from two models, likely further boosted performance, contributing to the SOTA results on SN-BAS.

4.6 PES ablations

In this section, we incrementally ablate the different components of our approach. We start with a base model consisting of the feature extractor without GSF, producing features with only spatial information, and plain layers of the different temporal approaches. Initially, we consider only the classification head (i.e., $\mathcal{L} = \mathcal{L}_c$), resulting in per-frame classifications. For each approach, we evaluate multiple configurations with different hyperparameters (e.g., number of layers) and present results with the best configuration, reporting the mAP with a tolerance of $\delta = 1$. Each experiment is trained with two different seeds and we report the average for robustness. Ablations are conducted using two datasets: FineDiving and FigureSkating on the FS-Comp split.

Promoting token discriminability. As previously discussed, tasks like PES require a large enough number of tokens for sufficient temporal precision. However, the greater the number of tokens the larger the information redundancy, especially between nearby tokens (corresponding to apperance-wise similar frames). Increasing the discriminability among those tokens is key to preventing precision loss of the predictions. Here, we evaluate the discriminability of three commonly used layers for modeling temporal information: Transformer [42], GRU [39], and SGP [9]. To quantify discriminability, we use the validation split, first scaling the tokens within each feature dimension (i.e., $z^l_{i,d} / \max_j |z^l_{j,d}|$, with $z^l \in \mathbb{R}^{K \times D}$ representing all K tokens at layer l, each of dimension D) to ensure all values fall within the same range, as justified in D. We then compute the average cosine similarity between each token and the mean token of the sequence, following Shi et al. [9].

Figure 5 illustrates this analysis. In the initial layers, we observe similar patterns between the Transformer and the SGP, where tokens exhibit high similarity after the spatial backbone(BB) and the addition of learnable positional encodings(PE). However, token similarity decreases after the introduction of the first layer that incorporates temporal context(L1), allowing for better differentiation between similar frames. As we move through the layers, the SGP maintains either similar or slightly increased similarity, whereas in the Transformer, the increase is more pronounced, particularly in the final layer. This result highlights the rank-loss problem in Transformer, leading to reduced token discriminability. In contrast, the GRU, which typically requires fewer layers, exhibits a different pattern: it shows lower similarity after the backbone and positional encoding but increased similarity in the output tokens, with values higher than those of the SGP. As a result, SGP layers present the lowest token similarity (i.e., the highest discriminability), proving their effectiveness



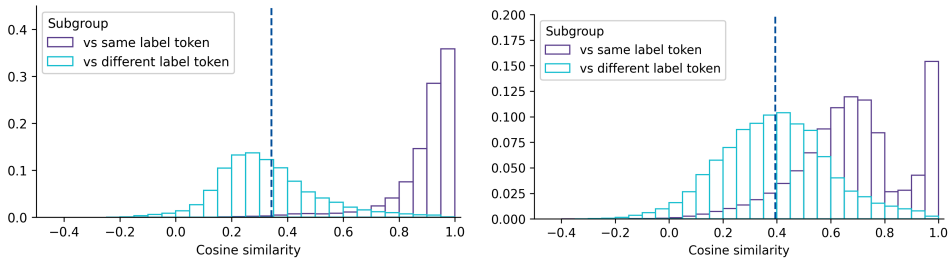
(a) FS-Comp discriminability analysis. (b) FineDiving discriminability analysis.

Fig. 5: **Temporal module discriminability analysis.** Cosine similarity after backbone (BB), post-positional encoding (PE), and at each temporal layer is displayed. Additionally, mAP performance with $\delta = 1$ is reported.

in enhancing token distinctiveness. This enhanced discriminability also translates to superior performance across temporal modules, as shown in Table 5(a). These findings highlight the benefits of using the SGP layer, especially for precision-demanding tasks like PES.

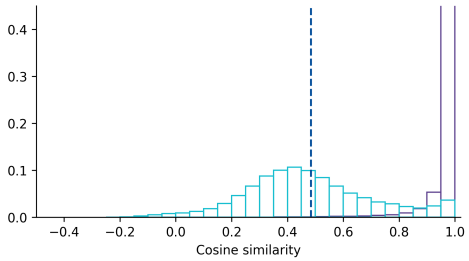
We further analyze the discriminability between two subgroups of tokens: those that, in the ground-truth space, correspond to the same event (e.g., two background positions or two temporal positions within the same event class), and those corresponding to different classes (e.g., a background token versus a token representing a specific event). This analysis is conducted on the last layer of the different modules, as we are particularly interested in achieving high discriminability just before the prediction heads—especially among tokens representing different events, rather than those from the same class. In addition, we incorporate the SGP-Mixer into this analysis, since the outputs on the final layer are comparable—unlike the previous per-layer analysis where different temporal resolutions were a factor.

As shown in Figure 6, we observe distinct distributions of token similarities between the two subgroups across all temporal modules. Tokens representing different actions exhibit lower similarity, which is a desirable result as it reflects an improved ability to distinguish between different action labels. This difference in similarity distribution is particularly pronounced in the FigureSkating dataset, leading to higher mAP scores, as shown in Table 5, while posing more challenges for the FineDiving dataset. Moreover, the findings from our previous analysis on overall discriminability align with this subgroup analysis. Specifically, SGP and SGP-Mixer architectures show lower similarity among tokens of different labels when compared to Transformers and GRUs, indicating superior discriminability. More importantly, SGP-Mixer further improves the discriminability in the FigureSkating dataset. In the FineDiving dataset, SGP certainly shows slightly better discriminability than SGP-Mixer. However, the latter seems to generate more distinct distributions between the two subgroups.

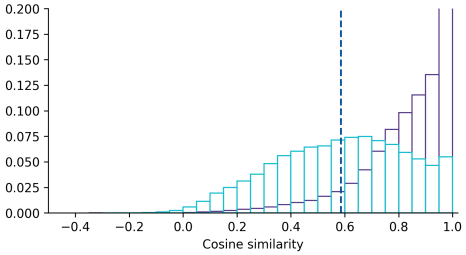


(a) FigureSkating - GRU

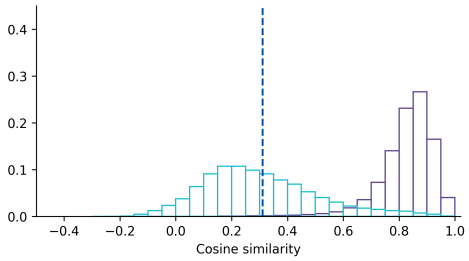
(b) FineDiving - GRU



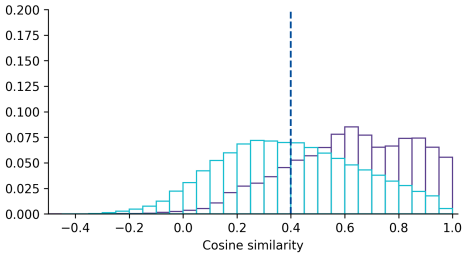
(c) FigureSkating - Transformer



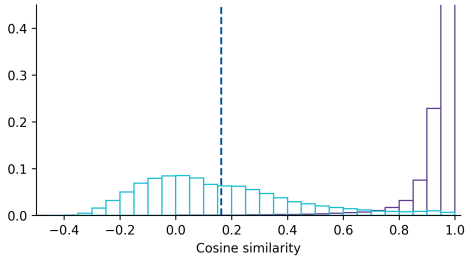
(d) FineDiving - Transformer



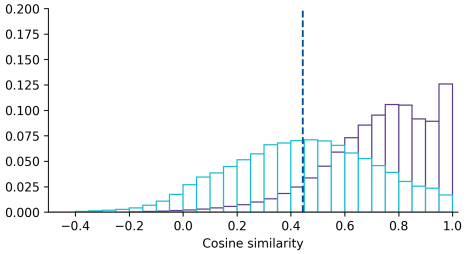
(e) FigureSkating - SGP



(f) FineDiving - SGP



(g) FigureSkating - SGP-Mixer



(h) FineDiving - SGP-Mixer

Fig. 6: Discriminability analysis in the last layer for different temporal modules: GRU, Transformer, SGP, and SGP-Mixer. The distributions of token similarity are shown for two subgroups: tokens with the same ground-truth label and those with different ground-truth labels. The vertical line indicates the mean similarity for the second subgroup (i.e., "vs different label token").

Experiments	FS-Comp mAP ($\delta = 1$)	FineDiving mAP ($\delta = 1$)
(a) Temporal module		
Transformer	69.68	59.84
GRU	72.04	59.09
SGP	**73.14**	**60.07**
(b) Skip connection		
w/o	74.29	63.01
sum	73.27	60.14
concat	76.78	61.90
SGP-Mixer (sum)	76.88	61.34
SGP-Mixer	**77.96**	**63.67**

Table 5: Ablation of T-DEED's main components using mAP with $\delta = 1$, highlighting the best results in bold.

Defining multiple temporal scales. Employing multiple temporal scales while processing videos has proven effective across various TAL and AS methodologies [9, 13, 32]. Here, we evaluate our encoder-decoder architecture, operating on multiple temporal scales, using diverse mixture approaches in skip connections. Specifically, we assess five variations of the encoder-decoder architecture: (1) no skip connections, (2) addition, (3) concatenation and linear projection, (4) a modified SGP-Mixer aggregating branches information with summation, and (5) our proposed SGP-Mixer layer. Results of these approaches are presented in Table 5b.

We observe that all encoder-decoder based approaches, with or without skip connections, outperform the baseline using a single scale (Transformer, GRU, and SGP). This underscores the importance of capturing information from multiple temporal scales, effectively achieved in our encoder-decoder. However, further analysis reveals issues with approaches utilizing addition for feature aggregation within skip connections, as they tend to yield inferior results compared to alternative methods. We hypothesize that this problem may be due to the attribution of equal weight to features from skip connections and previous layers when adding them. This could be particularly critical in the top layers of the architecture, where information passed from skip connections may be too primitive. Finally, our proposed SGP-Mixer layer stands as the best approach for both datasets in aggregating information within the skip connections. This emphasizes the advantages of our SGP-Mixer layer in aggregating information across multiple temporal scales while promoting token discriminability.

Introducing the displacement head. In Table 6a we evaluate various methods for addressing the imbalance between frames containing actual events and background frames in PES. We include label dilation, which involves extending the ground-truth positive labels by a specified radius around the exact temporal position during training, and may impact prediction precision. Additionally, we asses the utilization of a displacement head, as detailed in Section 3.3. Results indicate that the prediction head offers more benefits compared to label dilation, enabling wider range of event

detection without sacrificing prediction precision.

Experiments	FS-Comp mAP ($\delta = 1$)	FineDiving mAP ($\delta = 1$)
(a) Displacement head		
w/o & dilation = 0	74.74	63.67
w/o & dilation = 1	77.96	61.97
$r_E = 1$	**78.20**	67.49
$r_E = 2$	77.12	**68.31**
(b) Feature pyramids		
Tridet	68.31	64.28
(c) Feature extractor		
w/ gsm	81.00	67.95
w/ gsm (half)	81.05	67.47
w/ gsf	80.43	67.87
w/ gsf (half)	**81.25**	**68.40**
(d) Clip length		
$L = 25$ frames	71.02	66.61
$L = 50$ frames	76.87	64.84
$L = 100$ frames	**81.25**	**68.40**
$L = 200$ frames	79.19	65.29
(e) Postprocessing		
NMS [48]	83.79	71.10
SNMS [44]	**85.15**	**71.48**

Table 6: Further ablations and analysis of T-DEED using mAP with $\delta = 1$.

Feature pyramids. A common approach in TAL to process multiple temporal scales is the use of feature pyramids, which resembles using only the encoder of our approach. In Table 6b, we explore adapting the SGP Feature Pyramid proposed in Shi et al. [9] to our task. This method generates predictions at each temporal scale and integrates a displacement head to locate them. However, we observe a performance decrease compared to our encoder-decoder architecture. Further analysis of the predictions produced at each layer of the pyramid, depicted in Figure 7, reveals a decrease in performance as we descend the pyramid towards more high-level but lower-resolution features. This suggests that generating predictions at lower resolutions may compromise the accuracy. At the same time, predictions based on the low-level features on the initial layers are also suboptimal due to their lack of temporal context. Our encoder-decoder architecture is able to model high-level information while recovering the original temporal resolution, thus avoiding losing the temporal precision that is critical in PES.

Feature extractor ablations. In Table 6c, we explore the impact of integrating different temporal shift modules into our 2D backbone to capture local temporal context. Specifically, we consider two modules, GSM [38] and GSF [40], known for

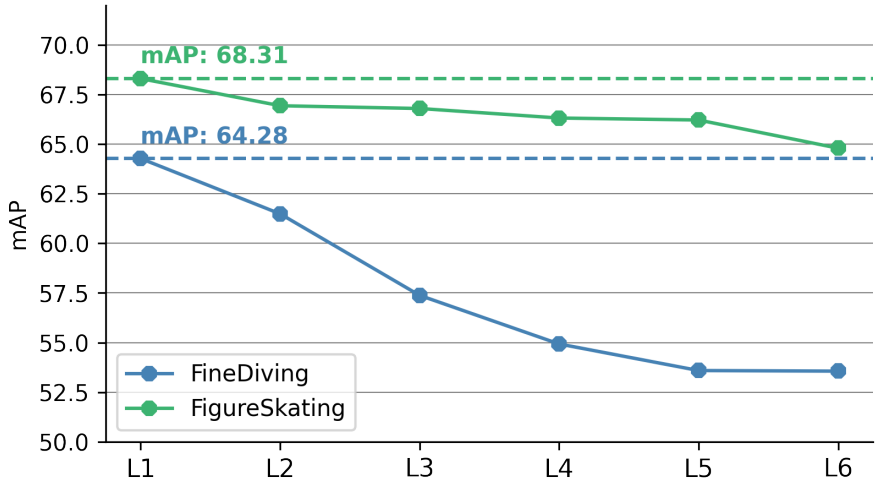


Fig. 7: Per-layer mAP analysis using the Feature Pyramid Network. mAP is reported at each layer of the SGP feature pyramid, accumulating predictions from previous layers, for the FineDiving and FigureSkating datasets.

their effectiveness in action recognition tasks. We assess two variants: one applying these modules across all backbone layers, and another limited to the latter half to promote stronger spatial modeling before temporal integration. Following [3], we apply these modules to $\frac{1}{4}$ of the channels of the residual blocks. Results indicate that incorporating local temporal modeling into the backbone is especially useful in the FigureSkating dataset, while FineDiving shows comparable results to those without temporal modules. Notably, GSF applied to the latter half of the backbone yields the best performance for both datasets.

Clip length analysis. Regarding optimal clip length, Table 6d evaluates various number of frames. In FigureSkating, performance improves with increasing clip length, plateauing at 100 frames. Similarly, results for FineDiving also indicate 100 frames as the optimal choice.

Postprocessing analysis. In Table 6e, we illustrate the impact of employing two different postprocessing techniques, Non-Maximum Suppression (NMS) [48] and Soft Non-Maximum Suppression (SNMS) [44], after adjusting their parameters. The optimal results are achieved with a 1-frame window for NMS and a 3-frame window for SNMS. Notably, SNMS consistently improves mAP for both datasets.

Action	Training data *BAS*	*BAS+AS*	Action	Training data *BAS*	*BAS+AS*
Pass	85.13	85.93$_{+0.80}$	Cross	69.46	75.84$_{+6.38}$
Drive	79.00	79.56$_{+0.56}$	Ball player block	20.77	28.87$_{+8.10}$
High pass	75.43	81.29$_{+5.86}$	Shot	46.68	70.07$_{+23.39}$
Header	66.64	67.05$_{+0.41}$	Successful tackle	2.50	1.21$_{-1.29}$
Ball out of play	23.63	29.19$_{+5.56}$	Free-kick	71.22	92.43$_{+21.21}$
Throw-in	71.19	83.54$_{+12.35}$	Goal	5.51	11.94$_{+6.43}$
mAP	51.43	58.91$_{+7.48}$			

Table 7: Per-class results of T-DEED on the SoccerNet BAS dataset, comparing the use of only SN-BAS data and the combination of SN-AS and SN-BAS data. Table shows the Average-AP score for each action, sorted by frequency from most to least common.

4.7 SN-BAS ablations

In addition to the previous ablations that justified the various components of T-DEED, this section focuses on the adapation made to fit T-DEED to the SN-BAS dataset. Specifically, we show the importance of complementing the training with the original SN-AS dataset by using two prediction heads, which help mitigate the problems of the small SN-BAS dataset. Unlike the results reported in the challenge, here we train T-DEED on the training splits of both SN-AS and SN-BAS, use the validation splits for early stopping, and evaluate only on the test split of SN-BAS.

Table 7 compares T-DEED trained on both SN-AS and SN-BAS to training only on SN-BAS. The results show that incorporating the additional related data from SN-AS leads to a mAP improvement of +7.48 points. Examining the per-class improvements, we observe that the classes benefiting most from joint training are those present in both datasets and few instances in SN-BAS, such as shots, free-kicks, and throw-ins. However, for classes with high intra-class variability and limited samples, such as ball player blocks and successful tackles, performance remains low, indicating the need for more data to improve detection of these actions.

4.8 Qualitative results

Finally, we present qualitative results for all datasets in Figures 8, 9, 10, 11, 12, 13, and 14. For a sample clip from the test split, we display ground-truth annotations alongside the predicted probabilities for each action class.

In Figures 8 and 9, the predictions on the FigureSkating dataset are highly aligned with the ground-truth annotations, with only minor temporal deviations and consistently high confidence. Similarly, Figure 10 shows strong performance on the FineDiving dataset, though some duplicated detections occur near a ground-truth *twist*, indicating challenges in precisely localizing the *twist* action, as detecting the exact moment when the rotation around the body's vertical axis finishes can be difficult, especially depending on certain camera angles. In the Tennis dataset, as shown

in Figure 11, the qualitative results are consistent with the quantitative findings, with a high level of alignment between predictions and ground truth. However, an incorrect prediction is observed for a *far court swing*, where a swing is detected after the point has ended, which should not have been the case. Figure 12 highlights the challenges posed by the FineGym dataset. Although T-DEED detects most actions, it does so with reduced confidence and multiple predictions around the ground-truth, demonstrating the difficulties in achieving precise predictions for some classes. Finally, for the SoccerNet datasets (SN-AS and SN-BAS), as seen in Figures 13 and 14, the results are good, with most ground-truth actions correctly detected. However, some imprecision is observed in a few predictions, but confidence remains high, especially near the ground-truth locations.

5 Conclusion

This work presented T-DEED, a model designed to address Precise Event Spotting across various sports datasets. Ablation studies underscored the importance of processing videos in multiple temporal scales and promoting token discriminability for precise predictions. To address these challenges, we integrated an encoder-decoder architecture and proposed the SGP-Mixer layer, aimed at aggregating information at various temporal scales within skip connections while improving token discriminability. In our experiments, T-DEED achieved SOTA performance on four different PES sports datasets (FigureSkating, FineDiving, Tennis, and FineGym), obtained the best results among methods not using pre-extracted features on SoccerNet Action Spotting, and secured 1st place in the 2024 SoccerNet Ball Action Spotting Challenge.

Declarations

Funding. This work has been partially supported by the Spanish project PID2022-136436NB-I00 and by ICREA under the ICREA Academia programme.

A Datasets per-class analysis

In this section, we report the number of observations for each event or action across all datasets. Additionally, for FigureSkating, FineDiving, Tennis, and FineGym, we provide the Average Precision obtained by our model with the two RegNetY's variants (200MF and 800MF) for the different classes separately, extending the analysis in Section 4.5. For SN-AS and SN-BAS we only show the number of observations per class, given that per-class results have already been analyzed in the main paper.

		AP ($\delta = 1$)			
		FS-Comp		*FS-Perf*	
Event	Nº observations	200MF	800MF	200MF	800MF
Jump takeoff	1464	86.21	85.66	88.39	90.83
Jump landing	1464	83.70	85.24	88.18	90.67
Flying spin takeoff	373	86.03	85.06	84.24	87.07
Flying spin landing	373	83.83	82.73	86.35	84.12

Table 8: Classes, number of observations per class, and per-class Average Precision with a tolerance of $\delta = 1$ frame in the FigureSkating dataset.

		AP ($\delta = 1$)	
Event	Nº observations	200MF	800MF
Entry	2984	72.70	76.70
Som(s).Pike	2152	75.87	75.76
Som(s).Tuck	1071	75.71	74.77
Twist(s)	803	61.87	65.70

Table 9: Classes, number of observations per class, and per-class Average Precision with a tolerance of $\delta = 1$ frame in the FineDiving dataset.

Event	Nº observations	AP ($\delta = 1$) 200MF	800MF
Far-court ball bounce (FCB)	8150	94.56	95.06
Near-court ball bounce (NCB)	8127	95.23	95.54
Far-court swing (FCSw)	7123	95.28	97.23
Near-court swing (NCSw)	7044	98.36	98.43
Near-court serve (NCSe)	1690	99.26	99.01
Far-court serve (FCSe)	1657	99.47	99.39

Table 10: Classes, number of observations per class, and per-class Average Precision with a tolerance of $\delta = 1$ frame in the Tennis dataset.

Event	Nº observations
Ball out	31810
Throw-in	18918
Foul	11674
Indirect FK	10521
Clearance	7896
Shot on target	5820
Shot off target	5256
Corner	4836
Substitution	2839
Kick-off	2566
Direct FK	2200
Offside	2098
Yellow Card (YC)	2047
Goal	1703
Penalty	173
Red Card (RC)	55
YC $\rightarrow$ RC	46

Table 11: Classes and number of observations per class in the SoccerNet Action Spotting dataset.

Event	Nº observations
Pass	4985
Drive	4300
High pass	761
Header	713
Ball out of play	551
Throw-in	362
Cross	261
Ball player block	223
Shot	169
Player successful tackle	74
Free-kick	11
Goal	12

Table 12: Classes and number of observations per class in the SoccerNet Ball Action Spotting dataset.

Event	Nº observations	AP ($\delta = 1$) 200MF	800MF
Uneven bars circles start	6612	42.95	43.83
Uneven bars circles end	6612	59.98	63.25
Balance beam leap_jump_hop start	4787	63.32	65.17
Balance beam leap_jump_hop end	4787	44.48	44.63
Balance beam flight_salto start	4187	64.60	65.23
Balance beam flight_salto end	4187	30.70	31.82
Uneven bars transition_flight start	3389	81.14	84.80
Uneven bars transition_flight end	3389	82.89	83.51
Floor exercise leap_jump_hop start	3238	80.29	80.79
Floor exercise leap_jump_hop end	3238	54.90	56.80
Floor exercise back_salto start	2978	90.37	90.42
Floor exercise back_salto end	2978	49.19	48.97
Balance beam flight_handspring start	2893	58.85	59.77
Balance beam flight_handspring end	2893	72.58	70.92
Vault (timestamp 0)	2031	11.83	11.00
Vault (timestamp 1)	2031	70.93	72.49
Vault (timestamp 2)	2031	90.70	90.29
Vault (timestamp 3)	2031	29.46	31.24
Uneven bars flight_same_bar start	1624	80.28	79.98
Uneven bars flight_same_bar end	1624	75.84	75.87
Balance beam turns start	1371	41.87	42.21
Balance beam turns end	1371	22.80	24.09
Floor exercise from_salto start	1345	76.66	76.49
Floor exercise from_salto end	1345	40.60	41.10
Uneven bars dismounts start	1227	89.24	89.92
Uneven bars dismounts end	1227	40.33	39.09
Balance beam dismounts start	1218	80.96	80.09
Balance beam dismounts end	1218	27.64	26.26
Floor exercise turns start	1103	42.46	39.31
Floor exercise turns end	1103	49.57	51.12
Floor exercise side_salto start	49	34.32	38.10
Floor exercise side_salto end	49	9.26	14.06

Table 13: Classes, number of observations per class, and per-class Average Precision with a tolerance of $\delta = 1$ frame in the FineGym dataset.

B Implementation details T-DEED

Here we outline the configuration used for each T-DEED model in the SOTA comparison from Table 1 in the main paper. All models apply data augmentations, including mixup with $\alpha = \beta = 0.2$, color jitter with probability 0.25, and Gaussian blur with probability 0.25. For FigureSkating and Tennis, frames of size 398×224 are randomly cropped to 224×224, while for FineDiving and FineGym, frames are resized to 224×224. In SN-AS and SN-BAS, frames are processed at the extracted resolution. The detection radius r_E is set to 4, 3, 2, 1, 1, and 0 for SN-BAS, SN-AS, FineDiving, FigureSkating, Tennis, and FineGym, respectively. Additionally, we apply a weight of $w = 5$ to the positive classes within the cross-entropy loss.

Among model-specific hyperparameters, we have the number of blocks (B), the kernel size (ks), and the scalable factor within the SGP module (r). These are chosen independently for all datasets:

- FS-Comps(T-DEED w/ 200MF): $B = 3$, $ks = 5$, $r = 2$.
- FS-Comp (T-DEED w/ 800MF): $B = 2$, $ks = 9$, $r = 4$.
- FS-Perf (T-DEED w/ 200MF): $B = 3$, $ks = 9$, $r = 2$.
- FS-Perf (T-DEED w/ 800MF): $B = 2$, $ks = 9$, $r = 4$.
- FineDiving (T-DEED w/ 200MF): $B = 2$, $ks = 7$, $r = 4$.
- FineDiving (T-DEED w/ 800MF): $B = 2$, $ks = 9$, $r = 4$.
- Tennis (T-DEED w/ 200MF): $B = 3$, $ks = 11$, $r = 2$.
- Tennis (T-DEED w/ 800MF): $B = 3$, $ks = 11$, $r = 4$.
- FineGym (T-DEED w/ 200MF): $B = 3$, $ks = 11$, $r = 4$.
- FineGym (T-DEED w/ 800MF): $B = 3$, $ks = 9$, $r = 4$.
- SN-AS (T-DEED w/ 200MF): $B = 3$, $ks = 9$, $r = 4$.
- SN-AS (T-DEED w/ 800MF): $B = 3$, $ks = 11$, $r = 4$.
- SN-BAS (T-DEED w/ 200MF): $B = 2$, $ks = 9$, $r = 4$.

C Qualitative results

In this section, we present qualitative results on a randomly selected clip from the test split of each dataset. We display both the ground-truth observations and T-DEED predictions to support the detailed analysis in Section 4.8. For datasets with a large number of classes, we omit classes that do not appear in the clip's ground truth for clarity in the visualizations.

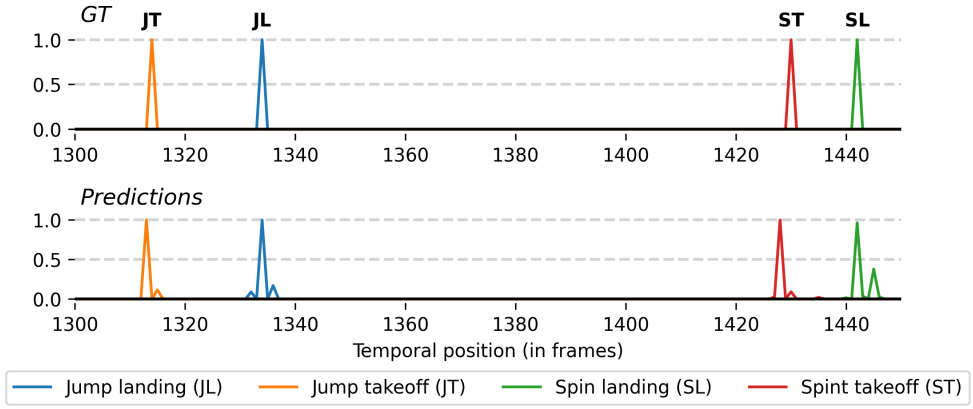


Fig. 8: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the FS-Comp split.

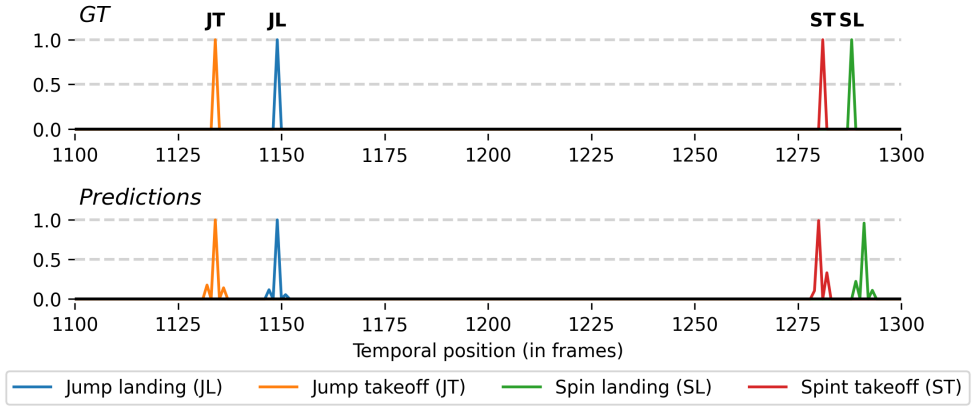


Fig. 9: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the FS-Perf split.

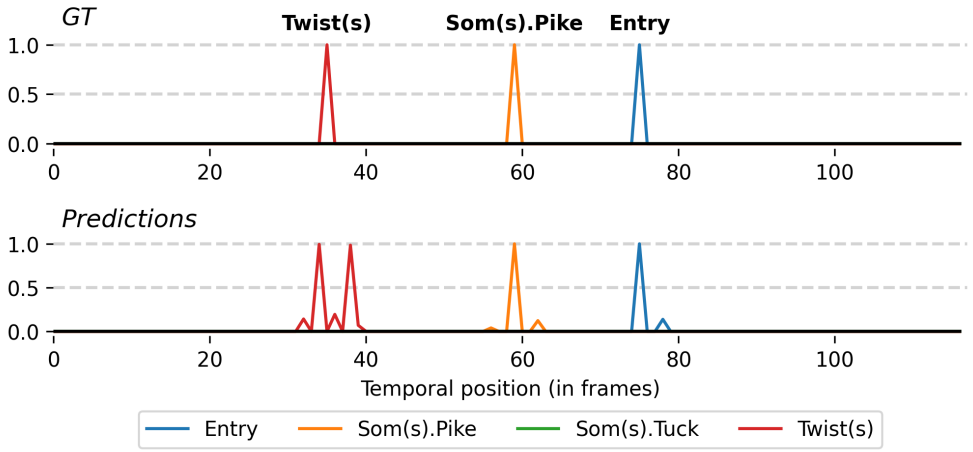


Fig. 10: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the FineDiving dataset.

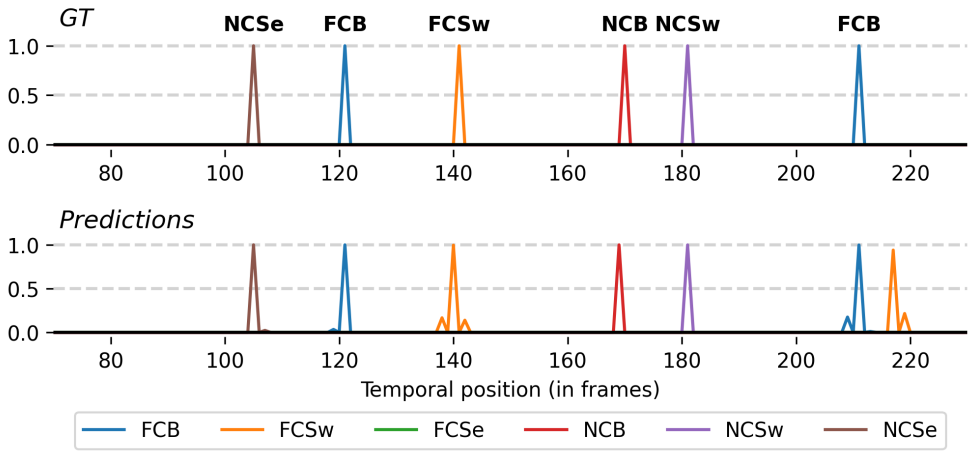


Fig. 11: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the Tennis dataset. Refer to Table 10 for the meanings of class label abbreviations.

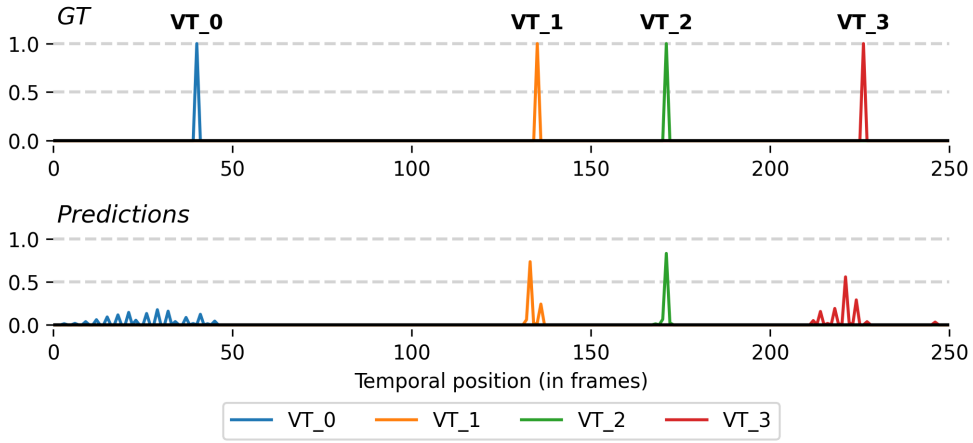


Fig. 12: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the FineGym dataset. Predictions of classes not present in the ground-truth were excluded for brevity.

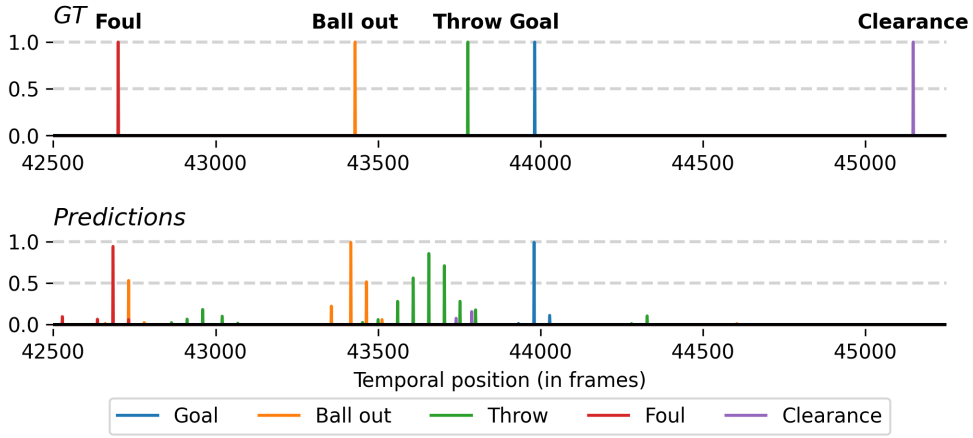


Fig. 13: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the SoccerNet Action Spotting dataset. Predictions of classes not present in the ground-truth were excluded for brevity.

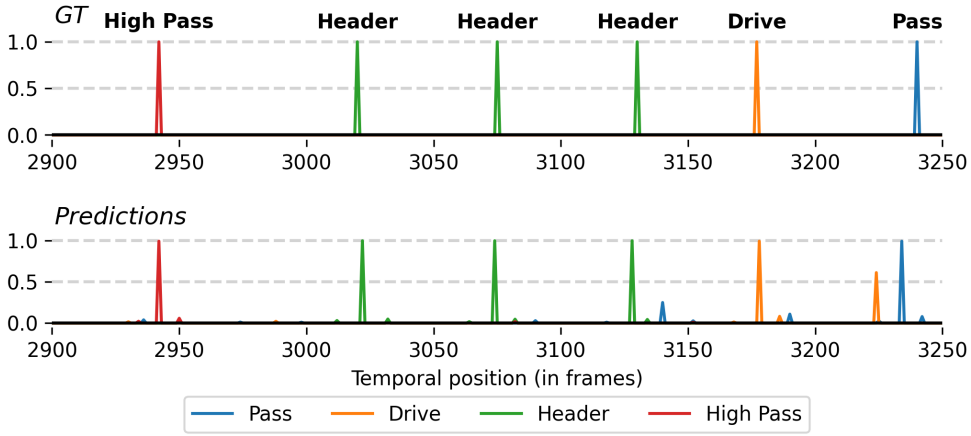
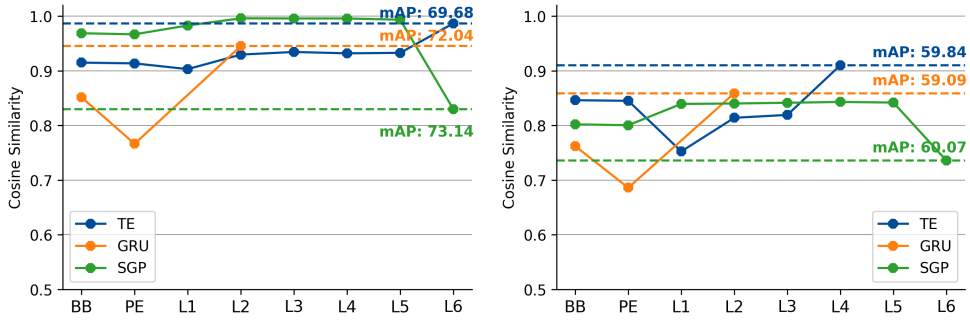


Fig. 14: Qualitative results from a sample clip in the test split, illustrating ground-truth annotations alongside predicted probabilities for each class at every temporal position in the SoccerNet Ball Action Spotting dataset. Predictions of classes not present in the ground-truth were excluded for brevity.

D Extended discriminability analysis

In this section, we present an additional analysis of discriminability to complement the findings in the main paper. Figure 15 displays the cosine similarity analysis without scaling the features. Notably, the patterns for both the Transformer and the GRU are quite similar, as the output tokens from these layers are already within a comparable range due to their inherent characteristics. In contrast, the pattern observed for the SGP layer differs. This discrepancy arises from the pre-norm setting of the SGP layer, which, despite containing normalization layers at the layer's entrance, does not constrain the output tokens to a limited range of values. As a result, feature dimensions vary widely in range, with some dimensions being substantially larger than others. This variability influences the computed cosine similarity, which assesses the angle between vectors; the dominant dimensions can overshadow those with lower ranges, thus distorting the metric.



(a) FS-Comp discriminability analysis.

(b) FineDiving discriminability analysis.

Fig. 15: **Temporal module discriminability analysis.** Cosine similarity after backbone (BB), post-positional encoding (PE), and at each temporal layer is displayed. Additionally, mAP performance with $\delta = 1$ is reported.

To address this issue, we previously scale the tokens for each feature dimension by dividing them by their respective maximum values across the validation data. This operation brings all dimension in a consistent range without altering their distribution, thereby allowing the detection of differences across all dimensions when computing the cosine similarity.

References

[1] Herath, S., Harandi, M., Porikli, F.: Going deeper into action recognition: A survey. Image and vision computing **60**, 4–21 (2017)

[2] Xia, H., Zhan, Y.: A survey on temporal action localization. IEEE Access **8**, 70477–70487 (2020)

[3] Hong, J., Zhang, H., Gharbi, M., Fisher, M., Fatahalian, K.: Spotting temporally precise, fine-grained events in video. In: European Conference on Computer Vision, pp. 33–51 (2022). Springer

[4] Xarles, A., Escalera, S., Moeslund, T.B., Clapés, A.: 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, pp. 3410–3419 (2024)

[5] Hong, J., Fisher, M., Gharbi, M., Fatahalian, K.: Video pose distillation for few-shot, fine-grained sports action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9254–9263 (2021)

[6] Xu, J., Rao, Y., Yu, X., Chen, G., Zhou, J., Lu, J.: Finediving: A fine-grained dataset for procedure-aware action quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2949–2958 (2022)

[7] Shao, D., Zhao, Y., Dai, B., Lin, D.: Finegym: A hierarchical video dataset for fine-grained action understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2616–2625 (2020)

[8] Zhang, H., Sciutto, C., Agrawala, M., Fatahalian, K.: Vid2player: Controllable video sprites that behave and appear like professional tennis players. ACM Transactions on Graphics (TOG) **40**(3), 1–16 (2021)

[9] Shi, D., Zhong, Y., Cao, Q., Ma, L., Li, J., Tao, D.: Tridet: Temporal action detection with relative boundary modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18857–18866 (2023)

[10] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)

[11] Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)

[12] Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., *et al.*: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5842–5850 (2017)

[13] Soares, J.V., Shah, A., Biswas, T.: Temporally precise action spotting in soccer videos using dense detection anchors. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 2796–2800 (2022). IEEE

[14] Xarles, A., Escalera, S., Moeslund, T.B., Clapés, A.: Astra: An action spotting transformer for soccer videos. In: Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports, pp. 93–102 (2023)

[15] Cioppa, A., Giancola, S., Somers, V., Magera, F., Zhou, X., Mkhallati, H., Deliege, A., Held, J., Hinojosa, C., Mansourian, A.M., et al.: Soccernet 2023 challenges results. arXiv preprint arXiv:2309.06006 (2023)

[16] Giancola, S., Cioppa, A., Deliège, A., Magera, F., Somers, V., Kang, L., Zhou, X., Barnich, O., De Vleeschouwer, C., Alahi, A., *et al.*: Soccernet 2022 challenges results. In: Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports, pp. 75–86 (2022)

[17] Denize, J., Liashuha, M., Rabarisoa, J., Orcesi, A., Hérault, R.: Comedian: Self-supervised learning and knowledge distillation for action spotting using transformers. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 530–540 (2024)

[18] Giancola, S., Amine, M., Dghaily, T., Ghanem, B.: Soccernet: A scalable dataset for action spotting in soccer videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1711–1721 (2018)

[19] Deliege, A., Cioppa, A., Giancola, S., Seikavandi, M.J., Dueholm, J.V., Nasrollahi, K., Ghanem, B., Moeslund, T.B., Van Droogenbroeck, M.: 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, pp. 4508–4519 (2021)

[20] Idrees, H., Zamir, A.R., Jiang, Y.-G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The thumos challenge on action recognition for videos "in the wild". Computer Vision and Image Understanding **155**, 1–23 (2017)

[21] Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition, pp. 961–970 (2015)

[22] Zhao, H., Torralba, A., Torresani, L., Yan, Z.: Hacs: Human action clips and segments dataset for recognition and temporal localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8668–8678 (2019)

[23] Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: Dense detailed labeling of actions in complex videos. International Journal of Computer Vision **126**, 375–389 (2018)

[24] Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., *et al.*: Scaling egocentric vision: The epic-kitchens dataset. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 720–736 (2018)

[25] Buch, S., Escorcia, V., Shen, C., Ghanem, B., Carlos Niebles, J.: Sst: Single-stream temporal action proposals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2911–2920 (2017)

[26] Escorcia, V., Caba Heilbron, F., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, pp. 768–784 (2016). Springer

[27] Heilbron, F.C., Niebles, J.C., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1914–1923 (2016)

[28] Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., Qiao, Y., Yan, J., Gao, C., Sang, N.: Temporal context aggregation network for temporal action proposal refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 485–494 (2021)

[29] Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10156–10165 (2020)

[30] Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3320–3329 (2021)

[31] Liu, X., Wang, Q., Hu, Y., Tang, X., Zhang, S., Bai, S., Bai, X.: End-to-end temporal action detection with transformer. IEEE Transactions on Image Processing **31**, 5427–5441 (2022)

[32] Zhang, C.-L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with

transformers. In: European Conference on Computer Vision, pp. 492–510 (2022). Springer

[33] Yang, L., Han, J., Zhao, T., Liu, N., Zhang, D.: Structured attention composition for temporal action localization. IEEE Transactions on Image Processing (2022)

[34] Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: Procedings of the British Machine Vision Conference 2017 (2019). British Machine Vision Association

[35] Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 988–996 (2017)

[36] Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE Transactions on Image Processing **29**, 8535–8548 (2020)

[37] Dong, Y., Cordonnier, J.-B., Loukas, A.: Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In: International Conference on Machine Learning, pp. 2793–2803 (2021). PMLR

[38] Sudhakaran, S., Escalera, S., Lanz, O.: Gate-shift networks for video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1102–1111 (2020)

[39] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

[40] Sudhakaran, S., Escalera, S., Lanz, O.: Gate-shift-fuse for video action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[41] Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10428–10436 (2020)

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[43] Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)

[44] Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5561–5569 (2017)

[45] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[46] Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981)

[47] Zhou, X., Kang, L., Cheng, Z., He, B., Xin, J.: Feature combination meets attention: Baidu soccer embeddings and transformer based temporal detection. arXiv preprint arXiv:2106.14447 (2021)

[48] Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, pp. 850–855 (2006). IEEE