Supplemental Material: Spotting Temporally Precise, Fine-Grained Events in Video

James Hong[1], Haotian Zhang[1], Michaël Gharbi[2],
Matthew Fisher[2], and Kayvon Fatahalian[1]

[1] Stanford University
[2] Adobe Research

A Implementation Details for E2E-Spot

A.1 Spatial-Temporal Feature Extractor, *F*

As described in § 3, our feature extractor is a standard RegNet-Y [19] with Gate Shift Modules [21] (GSM) inserted. GSM is applied at each residual block, to $\frac{1}{4}$ of the channels, rounded up to the nearest multiple of 4. RegNet-Y 200MF and 800MF produce spatially-pooled features of dimension 368 and 768 respectively.

We choose RegNet-Y [19] over the more commonly used ResNet [12] family of 2D CNNs because the former is more recent and compact (RegNet-Y 200MF has 3.2M parameters vs. 11.7M parameters for ResNet-18), while exhibiting generally better performance on image classification benchmarks [24]. E2E-Spot, however, can be implemented with any 2D CNN architecture.

A.2 Long-term Temporal Reasoning Module, *G*

G provides temporal reasoning on dense feature vectors, following the spatial pooling layer of F. The details of G are given in the paper in § 3.2. Here, we provide details for the additional variations of E2E-Spot used in § 5.4.

Deeper GRU increases the number of GRU layers to 3. *MS-TCN and AS-Former* are described in § B.1.

*GRU** takes multiple 1-layer GRUs at different temporal granularities, in addition to the 1-layer GRU, to more directly aggregate information across wider contexts. We use two temporal scales, 4 and 16, requiring two additional GRUs. Each scale defines a temporal down-sampling of the clip length by a factor of the scale, S. For each scale, all output features are first fed to a fully connected layer and ReLU. Then, the sequence of length N is divided into $\lceil \frac{N}{S} \rceil$ non-overlapping windows, and max-pooling is performed in each window. The $\lceil \frac{N}{S} \rceil$ sequence is processed by the scale-specific GRU. Finally, the outputs of each GRU, at each time scale, are up-sampled by repetition back to the full clip length N and concatenated for each time step t.

While these experiments do not cover the full breadth of architectures and settings available, we note that we did not observe major performance gains over the 1-layer GRU in applying these alternatives alongside end-to-end learning.

A.3 Training Configuration

We train E2E-Spot using 100 frame long clips by default and a batch size of 8 clips. Batches are formed by randomly sampling clips from the training videos. We group every 625 training steps into a training cycle (i.e., a pseudo-epoch of 500K frames). A single cycle runs in approximately 8.5 and 14 minutes on a single A5000 GPU [17] for the 200MF and 800MF variants, respectively. All variations of E2E-Spot are trained for 50 cycles on the Tennis, Figure Skating, and FineDiving datasets. We train the 200MF model for 100 cycles and the 800MF model for 150 cycles on FineGym and SoccerNet-v2, due to the larger dataset sizes (see § D). Training is performed with AdamW [16], setting a base learning rate of 10^{-3}, with 3 linear warmup cycles followed by cosine decay [15].

Data Augmentations. We randomly apply color jitter, Gaussian blur, and mixup [29] during training. On Tennis, Figure Skating, and FineGym, we also randomly crop the 398×224 frames to 224×224 pixels. This crop only affects the width dimension, as cropping the height dimension can lead to precise events falling outside the visible field (e.g., the tennis court and player span the vertical dimension). For FineDiving [25], we use the frames extracted by the original authors (256 pixels in the vertical dimension) and random crops of 224×224 pixels. Finally, for SoccerNet-v2, we do not use random cropping because context such as the goal or the field boundary are often at the periphery of the frame.

For Figure Skating only (FS-Perf and FS-Comp), we use label dilation of ± 1 frames due to the very large imbalance between events and background frames (see § D.2). Label dilation is beneficial on Figure Skating for both E2E-Spot and the baselines (see § C.2). Note that label dilation is not used during testing.

Non-maximum Suppression. We evaluated the model predictions with and without non-maximum suppression (NMS). For the temporally precise datasets, we used a window of ± 1 frames whereas we use ± 2 frames at 2 FPS for SoccerNet-v2. The efficacy of NMS in the temporally precise setting depends on the frame level tolerance, dataset, and model (see experiments in § C.3), so the decision to apply NMS in practice should be made with application and task requirements in mind.

A.4 Optical Flow Extraction, for Additional Experiments

We use optical flow extracted by RAFT [22] for the additional 2-stream experiments that we described in § 5.4. During preprocessing, we subtract the median flow value for each frame and clamp to a range of $[-20, +20]$ pixels.

B Implementation Details for Baselines

We adapt a number of published architectures from the action segmentation (TAS), detection (TAD), and spotting literature as baselines for temporally precise spotting and provide their key implementation details here.

B.1 Models

TCN and MS-TCN. We adapt the code from Farha et al. [9], using dilated temporal convolution networks. Multiple stages typically improves results over a single stage TCN. We use 3 TCN stages for our MS-TCN baselines and a depth of 5 layers for each stage. Each layer has dimension of 256. Per-frame predictions are made with a fully connected layer that maps from 256 to $K + 1$.

GRU. We use a bidirectional GRU [3] with 5 layers and a dimension H of 128. Per-frame predictions are made with a fully connected layer, from $2H$ to $K + 1$.

ASFormer. We use code and settings from the implementation by Yi et al. [27].

GCN. We use the GCNeXt block architecture proposed by Xu et al. [26], which produces a 256 dimensional feature encoding for each frame. Per-frame predictions are made with a fully connected layer mapping from 256 to $K + 1$.

StridedTransformer. We implement a transformer [18] that operates on a window of per-frame features [30]. The model takes a consecutive clip of 31 features and positional encodings, and it predicts whether the center frame is one of the K events or not.

NetVLAD++ [10] is used similarly to the transformer described above. We observe on precise spotting tasks that NetVLAD++ often fails to overcome the class imbalance between foreground events and background frames. Reducing window size from 31 to 7 frames improves performance slightly, but overall performance remains poor and the StridedTransformer described above performs significantly better (see § C.1).

VC-Spot is a end-to-end learned video classification baseline, which, given a clip of 15 consecutive RGB frames, predicts whether the middle frame is an event. We use the same RegNet-Y 200MF (with GSM) CNN backbone as E2E-Spot. Training VC-Spot using batches containing randomly sampled clips fails to overcome the large foreground / background frame imbalance. This is a challenging problem since a window that contains a temporally precise event as its middle frame differs from its neighbors by only one frame in time. To ameliorate this, we form batches with densely overlapped clips (4 sequentially) in addition to the batch size of 8.

B.2 Pre-trained Features

We test I3D [2] and MViT base (MViT-B) [8] features trained on Kinetics-400 [14], without fine-tuning. I3D features are extracted following the example

of Farha et al. [9], with RGB and flow. MViT-B features use the 16x4 model in PyTorchVideo [7]. Performance with these features is poor — far below fine-tuned features such as TSP [1] (see Table C1 and C2). Due to the high cost of feature extraction on large datasets with I3D and the poor spotting performance of downstream models trained using I3D features, we only extract MViT-B [8] features for FineDiving and FineGym.

B.3 Fine-tuned Features

We test two fine-tuning strategies that use video clip classification in the target domain (i.e., the precise spotting dataset) as a fine-tuning step for temporal localization tasks.

Temporally Sensitive Pretraining (TSP). We use code from Alwassel et al. [1], which pre-trains a R(2+1)D-34 [23] model to encode spatial-temporal features. The model is first initialized with weights from a model trained on Kinetics-400 [14]. During fine-tuning, we use a clip length of 12 frames. For the pre-trained global video feature (GVF), we use pre-extracted MViT-B [8] features (from § B.2) as these serve a similar function to the frozen GVF in the original implementation. We optimize the model using TSP until its validation loss and accuracy converges.

(K + 1)-VC pre-trains a RegNet-Y 200MF with GSM on a standard video classification task. It is included to demonstrate a simpler fine-tuning baseline than TSP, using a feature extractor of comparable complexity and architecture to the one that we selected for E2E-Spot.

We initialize the RegNet-Y backbone with pre-trained weights learned on ImageNet-1K [6]. For fine-tuning, we use a clip length of 7 frames. A small clip length is selected because the goal is to learn a localized, per-frame feature; downstream models for spotting will receive a long sequence of these features. Clips of the K foreground classes contain a foreground event within a half clip length window in the clip center while background class clips do not. We sample background clips randomly with 20% probability during training. The model is trained with a batch size of 16 clips and for 18.8K steps. The best epoch is selected using validation accuracy.

Video Pose Distillation (VPD) [13] features are available for the Figure Skating dataset and serve as a strong baseline / performance target for E2E-Spot.

The VPD features are learned in an unsupervised manner over the entire video dataset (including the test videos, without access to action or event labels). They make use of hand-engineered subject tracking, RGB pixels, and optical flow as inputs. We test both 2D-VPD and (view-invariant) VI-VPD features. The differences are subtle when applied to precise spotting, with 2D-VPD being better a majority of the time (see § C.1).

B.4 Training Configuration (for Spotting)

With the exception of VC-Spot (an end-to-end learned baseline), all of the baseline architectures described in § B.1 operate in two phases, learning a spotting head on densely pre-extracted features.

We train the TCN, MS-TCN, GRU, ASFormer, and GCN models on randomly sampled, 500 frame long clips — with a batch size of 50, a train-val cycle of 40 steps (1M frames), and for 50 cycles. Updates are performed using AdamW [16] with a base learning rate of 10^{-3}, linear warmup (3 cycles), and cosine annealing [15]. The StridedTransformer and NetVLAD++ [10] baselines make singular predictions on a window of frames. We train these with a batch size of 100 clips, train-val cycles of 1,000 steps, and for 50 cycles. We use the same AdamW [16] optimizer and LR schedule as the other models. Validation mAP, computed at the end of every training cycle, is used for model selection.

C Additional Experiments & Ablations

In § C.1 and § C.2, we present additional baselines omitted from the main paper due to space constraints. § C.3 assesses the necessity of non-maximum suppression (NMS) for temporally precise spotting. § C.4 provides results when evaluating spotting performance at tolerance $\delta = 0$ frames (i.e., the exact frame of human annotation). § C.5 analyzes the variation in precise spotting performance among the event classes in each dataset.

C.1 Full Baseline Result Tables

We report the top baseline results in the main paper § 5.1. Table C1, C2, and C3 provide full results for all of the baselines and feature combinations.

For the best performing MS-TCN [9], GRU [3], and ASFormer [27] configurations, we further trained the model with and without CALF [4] and label dilation (propagating labels to ± 1 adjacent frames). NetVLAD++ [10] failed to overcome label sparsity in all tested datasets except for Tennis (with fine-tuned features). The StridedTransformer [18] performed better than NetVLAD++ and was tested with and without label dilation (± 1 frames), as it also suffers from sparsity in the foreground labels.

C.2 Impact of Additional Losses on Baseline Performance

Losses such as CALF [4] have been proposed in spotting literature as a way to address sparsity in temporal event labels. In the interest of obtaining strong baselines for precise spotting, we attempt to boost the top performing model architecture and feature baselines in § C.1.

We add CALF as an additional loss, with parameters that smooth around a event within a 7 frame window. Conceptually, because of the tight tolerances in temporally precise spotting, the small number of frames in an appropriately sized

Table C1: **Spotting performance (mAP @ δ frames)** using pre-trained features without fine-tuning. † indicates NMS. The best baseline scores are underlined. Due to the low performance of I3D [2] features (compared to TSP [1]), we do not extract I3D features for FineDiving and FineGym.

			Tennis		FS-Comp		FS-Perf		FineDiving		FineGym			
											Full		Start	
			δ=1	2	1	2	1	2	1	2	1	2	1	2
Default: E2E-Spot 200MF (RGB)			96.1	†97.7	†81.0	†93.5	†**85.1**	†95.7	**68.4**	†**85.3**	†47.9	†65.2	†61.0	†78.4
Best: E2E-Spot 800MF (2-stream)			†**96.9**	†**98.1**	†**83.4**	†**94.9**	†83.3	†**96.0**	†66.4	†84.8	†**51.8**	†**68.5**	†**65.3**	†**81.6**
Feature	Model	Extra loss (if any)												
I3D [2] (RGB + flow)	MS-TCN		62.7	75.0	60.8	†79.1	64.0	†83.6	-	-	-	-	-	-
	MS-TCN	CALF	59.7	73.6	56.4	†72.2	61.6	†81.5	-	-	-	-	-	-
	MS-TCN	dilate 1	58.1	†75.4	59.7	†79.5	69.0	†89.3	-	-	-	-	-	-
	GRU		40.7	†66.1	38.6	†58.7	41.4	†64.2	-	-	-	-	-	-
	GRU	CALF	†45.7	†70.5	†31.2	†53.0	†50.5	†75.4	-	-	-	-	-	-
	GRU	dilate 1	†41.5	†68.2	41.8	†69.8	52.5	†77.5	-	-	-	-	-	-
	ASFormer		55.4	†74.5	60.8	†82.2	69.0	†88.8	-	-	-	-	-	-
	ASFormer	CALF	58.1	†76.5	61.2	†82.4	66.6	†89.7	-	-	-	-	-	-
	ASFormer	dilate 1	49.6	†72.9	58.1	†81.1	64.6	†87.5	-	-	-	-	-	-
	TCN		†58.9	†75.1	†53.0	†72.0	†58.7	†81.3	-	-	-	-	-	-
	GCN		†42.6	†55.2	†19.9	†32.5	†27.1	†45.5	-	-	-	-	-	-
	StridedTF		†34.3	†48.0	†27.0	†43.8	†40.5	†63.6	-	-	-	-	-	-
	StridedTF	dilate 1	†44.8	†62.9	†36.2	†56.2	†47.2	†68.9	-	-	-	-	-	-
MViT-B [8] (RGB)	MS-TCN		67.0	†78.3	56.9	†75.8	63.6	†80.8	56.1	†73.9	31.0	†48.2	†41.7	†63.2
	MS-TCN	CALF	66.8	†79.3	57.4	†75.8	64.8	†84.3	56.3	†75.5	30.0	†48.3	40.1	†63.0
	MS-TCN	dilate 1	64.0	†80.1	55.6	†79.9	62.1	†82.9	59.3	†78.3	28.7	†48.6	†40.5	†64.8
	GRU		64.8	79.6	45.6	†69.6	56.8	†76.1	57.3	76.7	†25.9	†42.1	†34.0	†54.3
	GRU	CALF	59.1	†76.4	†45.5	†71.1	52.9	†77.3	55.8	75.6	†20.1	†34.4	†27.0	†45.3
	GRU	dilate 1	†61.4	†80.8	44.7	†73.1	55.1	†79.1	48.7	†76.5	†28.5	†48.6	†39.1	†62.2
	ASFormer		63.2	†79.9	55.8	†81.5	54.9	†80.4	37.4	†67.1	†24.9	†42.5	†32.4	†52.9
	ASFormer	CALF	63.9	†79.5	52.3	†76.6	55.7	†81.7	38.5	†67.4	†25.3	†42.9	†32.3	†53.8
	ASFormer	dilate 1	58.0	†78.9	†53.9	†81.8	56.4	†79.9	†35.2	†65.5	†23.4	†42.1	†32.5	†55.3
	TCN		†66.1	†80.4	†47.8	†67.9	†59.6	†80.2	†55.5	†77.2	†31.4	†49.1	†40.7	†62.8
	GCN		†36.4	†54.0	†20.8	†34.9	†27.7	†45.8	†38.8	†59.9	†12.3	†22.0	†16.8	†29.3
	StridedTF		†37.9	†54.9	†27.3	†45.7	†8.7	†15.2	†38.3	†64.7	†15.8	†25.4	†22.0	†34.3
	StridedTF	dilate 1	†54.8	†73.0	†32.0	†50.7	†39.7	†59.7	†42.1	†68.6	†20.6	†35.8	†26.4	†45.6

Table C2: **Spotting performance (mAP @ δ frames)** with features fine-tuned on RGB inputs. † indicates NMS. The best baseline scores are underlined.

			Tennis		FS-Comp		FS-Perf		FineDiving		FineGym			
											Full		Start	
			δ=1	2	1	2	1	2	1	2	1	2	1	2
Default: E2E-Spot 200MF (RGB)			96.1	†97.7	†81.0	†93.5	†**85.1**	†95.7	**68.4**	†**85.3**	†47.9	†65.2	†61.0	†78.4
Best: E2E-Spot 800MF (2-stream)			†**96.9**	†**98.1**	†**83.4**	†**94.9**	†83.3	†**96.0**	†66.4	†84.8	†**51.8**	†**68.5**	†**65.3**	†**81.6**
Feature	Model	Extra loss (if any)												
TSP [1]	MS-TCN		90.1	†94.6	72.4	†87.4	74.3	†89.4	55.5	†76.0	†40.5	†58.5	†53.9	†73.4
	MS-TCN	CALF	90.9	†95.0	72.1	†87.8	76.8	89.9	54.2	†73.8	36.9	†57.4	47.5	†71.4
	MS-TCN	dilate 1	†87.5	†95.1	67.0	†85.5	76.6	†89.3	57.7	†75.9	†37.8	†57.3	†53.2	†73.5
	GRU		89.5	95.1	66.6	†83.9	75.5	†89.4	55.5	76.5	†38.4	†57.2	†49.8	†70.5
	GRU	CALF	88.6	†94.9	64.4	†83.0	†60.1	†84.3	57.0	78.2	†36.1	†57.2	†44.3	†70.0
	GRU	dilate 1	†89.3	†96.0	†68.4	†88.3	†69.6	†90.6	†53.2	†77.4	†38.7	†58.8	†53.2	†74.2
	ASFormer		89.8	†94.8	77.7	†94.1	80.2	†94.5	47.1	†73.2	†38.8	†57.6	†51.1	†72.0
	ASFormer	CALF	89.0	†95.5	73.4	†92.5	78.0	†94.2	51.3	†77.4	†38.6	†57.6	†50.3	†71.6
	ASFormer	dilate 1	†86.9	†95.4	†72.2	†94.0	78.0	†94.0	†49.2	†76.4	†36.5	†57.6	†50.4	†72.9
	TCN		†88.1	†94.5	†62.6	†79.0	†67.3	†86.2	†51.9	†75.7	†41.1	†59.6	†53.5	†73.7
	GCN		†85.7	†93.4	†52.9	†70.6	†53.5	†74.8	†48.9	†71.0	†33.2	†49.5	†43.3	†62.2
	NetVLAD++		†55.5	†72.7	-	-	-	-	-	-	-	-	-	-
	StridedTF		†83.0	†93.3	†53.8	†73.3	†55.3	†76.9	†46.7	†74.2	†31.5	†47.8	†42.6	†60.9
	StridedTF	dilate 1	†86.0	†94.7	†61.2	†83.1	†65.3	†84.6	†46.6	†76.2	†31.7	†51.6	†39.6	†63.2
$(K+1)$-VC	MS-TCN		91.1	†94.8	66.5	†77.2	73.6	†83.8	63.2	†81.4	†40.9	†57.9	†53.2	†71.9
	MS-TCN	CALF	91.0	†94.5	60.8	†73.1	75.2	†86.7	59.0	†76.4	†38.6	†56.8	†50.1	†70.8
	MS-TCN	dilate 1	†90.3	†95.1	60.3	†73.6	77.2	†89.9	60.4	†83.5	†39.2	†58.2	†53.1	†73.8
	GRU		†90.8	†96.0	†61.1	†75.5	73.0	†86.5	60.0	†80.6	†41.1	†57.9	†54.3	†72.3
	GRU	CALF	†88.2	†95.4	†62.4	†77.2	†73.3	†85.0	61.8	†80.5	†39.6	†55.3	†51.8	†69.5
	GRU	dilate 1	†91.5	†96.2	†61.7	†78.9	†76.8	†89.4	†58.2	†82.6	†38.6	†57.5	†53.6	†73.6
	ASFormer		92.1	†95.5	67.2	†79.0	77.1	†88.9	†56.9	†83.0	†40.0	†56.8	†52.4	†70.3
	ASFormer	CALF	90.8	†94.5	67.6	†79.5	75.2	†88.3	58.9	†82.2	†40.0	†56.9	†52.9	†71.2
	ASFormer	dilate 1	†91.6	†96.2	†65.5	†79.8	75.4	†89.8	58.8	†83.5	†38.1	†56.9	†53.6	†72.9
	TCN		†91.9	†96.1	†58.8	†74.2	†74.5	†86.8	†58.6	†77.9	†42.0	†58.9	†54.6	†73.3
	GCN		†88.4	†94.2	†54.8	†68.0	72.6	†84.1	†55.3	†75.4	†32.6	†46.0	†43.2	†58.6
	NetVLAD++		†18.2	†26.3	-	-	-	-	-	-	-	-	-	-
	StridedTF		†88.4	†94.2	†39.2	†61.2	†59.1	†78.2	†50.4	†75.6	†24.7	†36.5	†34.4	†48.2
	StridedTF	dilate 1	†88.6	†95.2	†59.3	†77.2	†71.7	†87.6	†45.5	†75.3	†24.3	†39.2	†30.4	†48.3

Table C3: **Spotting performance (mAP @ δ frames) on FS-Comp and FS-Perf** using pose features [13], fine-tuned on RGB and optical flow. † indicates NMS. SOTA results with pose features are **bold**.

			FS-Comp		FS-Perf	
			δ=1	2	1	2
Default: E2E-Spot 200MF (RGB)			†81.0	†93.5	†85.1	†95.7
Best: E2E-Spot 800MF (2-stream)			†83.4	†94.9	†83.3	†96.0
Feature	Model	Extra loss (if any)				
2D-VPD [13]	MS-TCN		77.2	†90.8	83.1	†94.5
	MS-TCN	CALF	**83.5**	†**96.2**	**85.2**	†96.0
	MS-TCN	dilate 1	81.7	†95.5	82.4	†**96.4**
	GRU		†74.4	†94.2	†77.4	†94.9
	GRU	CALF	†72.2	†93.4	†46.3	†63.0
	GRU	dilate 1	†75.9	†94.3	†75.7	†94.1
	ASFormer		78.8	†94.8	76.9	†95.1
	ASFormer	CALF	78.2	†94.5	77.2	†93.9
	ASFormer	dilate 1	†79.0	†95.7	79.3	†93.2
	TCN		†75.0	†89.5	†76.5	†89.7
	GCN		†60.3	†72.5	†64.1	†77.2
	StridedTF		†12.7	†20.0	†26.0	†37.1
	StridedTF	dilate 1	†61.3	†79.2	†66.6	†84.2
VI-VPD [13]	MS-TCN		73.4	†88.8	80.8	†91.9
	MS-TCN	CALF	74.3	88.2	79.4	†91.3
	MS-TCN	dilate 1	77.8	†91.3	77.9	†92.7
	GRU		76.0	†94.8	78.2	†94.2
	GRU	CALF	†74.6	†93.7	†77.6	†93.5
	GRU	dilate 1	†74.9	†93.9	†77.6	†95.3
	ASFormer		77.4	†94.8	**85.2**	†95.6
	ASFormer	CALF	80.2	†94.5	84.2	†95.9
	ASFormer	dilate 1	79.7	†95.1	80.9	†93.7
	TCN		†68.3	†85.2	†73.9	†87.9
	GCN		†57.5	†71.6	†60.3	†71.6
	StridedTF		†23.4	†35.0	†67.5	†82.0
	StridedTF	dilate 1	†65.7	†82.3	†69.7	†87.7

Table C4: **Ablation of non-maximum suppression (NMS)** at different tolerances δ for various model and feature configurations. Best results per configuration are underlined. A spotting method's sensitivity to NMS can depend on the model (single vs. 2-stream), dataset, and feature type. The differences between NMS windows of 1 to 4 are also subtle, and a NMS window of 1 frame or none at all is often sufficient.

	Tennis		FS-Comp		FS-Perf		FineDiving		FineGym			
									Full		Start	
	δ=1	2	1	2	1	2	1	2	1	2	1	2
Default: E2E-Spot 200MF (RGB)												
No NMS	96.1	96.8	56.2	58.9	62.6	65.4	68.4	84.9	40.6	45.4	51.9	57.3
window = 1	96.1	96.7	81.0	93.5	85.1	95.7	66.3	85.3	47.9	65.2	61.0	78.4
window = 2	95.9	97.6	81.3	93.9	84.2	95.2	62.1	83.9	47.4	64.8	60.5	78.1
window = 4	95.7	97.4	81.2	93.8	84.1	95.1	59.2	81.6	47.0	64.2	60.2	77.6
Best: E2E-Spot 800MF (2-stream)												
No NMS	93.6	94.2	55.6	58.1	57.3	60.4	66.1	80.8	43.2	48.1	55.3	60.8
window = 1	96.9	98.1	83.4	94.9	83.3	96.0	66.4	84.7	51.8	68.5	65.3	81.6
window = 2	96.7	98.1	82.8	94.9	83.0	95.8	62.5	83.1	51.2	68.0	64.9	81.3
window = 4	96.6	98.0	82.8	94.9	83.0	95.8	59.9	81.0	50.7	67.2	64.6	80.9
Baseline: MS-TCN w/ TSP features												
No NMS	90.1	94.4	72.4	83.9	74.3	89.2	55.5	72.7	40.0	47.6	51.9	60.5
window = 1	87.6	94.6	68.2	87.4	68.1	89.4	50.9	76.0	40.5	58.5	53.9	73.4
window = 2	87.3	94.4	68.2	87.3	68.1	89.4	49.3	75.2	40.5	58.5	54.1	73.6
window = 4	87.0	94.0	68.2	87.3	68.1	89.4	47.7	73.4	40.4	58.3	54.0	73.4
Baseline: ASFormer w/ TSP features												
No NMS	92.1	94.0	67.2	75.3	77.1	85.9	56.8	69.5	33.8	39.1	42.9	48.5
window = 1	91.8	95.5	66.1	79.0	74.5	88.9	56.9	83.0	40.0	56.8	52.4	70.3
window = 2	91.5	95.4	66.1	79.0	74.5	88.9	55.9	82.3	39.9	56.7	52.3	70.3
window = 4	91.4	95.2	66.1	79.0	74.5	88.9	55.1	81.0	39.7	56.5	52.2	70.2

Table C5: **Spotting performance (mAP @ $\delta = 0$)**, when predicting the exact frame of human annotation. SOTA is **bold**. Best results per-category are otherwise underlined. As noted in § C.4, the conclusions that can be drawn from this table are limited because of ambiguity in the frame-level annotations.

	Tennis	FS-Comp	FS-Perf	FineDiving	FineGym	
					Full	Start
Default: E2E-Spot 200MF (RGB)	**71.6**	36.7	**40.5**	30.1	22.4	27.5
Best: E2E-Spot 800MF (2-stream)	69.1	37.6	38.6	**30.2**	**23.7**	**29.2**
MS-TCN w/ TSP features	50.0	33.3	34.0	23.2	19.7	25.3
w/ $(K+1)$-VC features	61.0	31.9	36.7	26.7	19.2	24.0
w/ 2D-VPD features & CALF	-	**43.1**	38.9	-	-	-
GRU w/ TSP features	42.4	30.6	27.4	15.4	18.6	23.5
w/ $(K+1)$-VC features	56.2	28.3	36.3	18.7	19.2	24.5
w/ 2D-VPD features & CALF	-	32.8	20.3	-	-	-
ASFormer w/ TSP features	51.6	36.6	37.4	22.5	18.6	23.6
w/ $(K+1)$-VC features	62.8	31.8	36.6	27.4	18.6	23.5
w/ 2D-VPD features & CALF	-	35.4	35.1	-	-	-

window prevents the loss from achieving as smooth as an effect as in coarse action spotting. We also implemented a simpler label dilation baseline, which addresses the sparsity problem by propagating event labels to ± 1 frame before and after each event at training time (denoted as "dilate 1").

Table C1, C2, and C3 list results with CALF and label dilation for the MS-TCN [9], GRU [3], and ASFormer [27] architectures. The results are generally mixed, with scores being similar with and without these loss modifications (e.g., within 1-2 mAP @ $\delta = 1$). On FS-Comp, the difference is more pronounced with 2D-VPD [13] features — up to 6.3 mAP improvement.

C.3 Sensitivity of Results to Non-Maximum Suppression

Non-maximum suppression (NMS) is a common post-processing technique in detection tasks [5,11]. We find that, for precise spotting, NMS is typically beneficial at tolerances of $\delta \geq 2$ frames but may be harmful for $\delta \leq 1$ frame (see Table C4). Tuning the NMS window threshold past 1 frame often has a minimal effect of less than 1 mAP point.

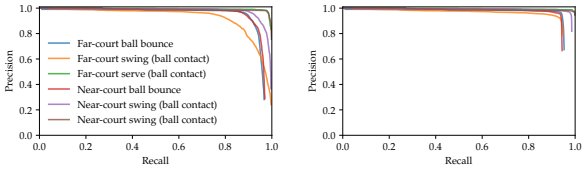
C.4 Predicting the Exact Frame of Human Annotation

While our spotting datasets have annotations at the frame-level, the $\delta = 0$ frame-prediction task is especially challenging to scientifically evaluate. In 25–30 FPS video, quick events such as a "ball bounce" can fall between two adjacent frames. $\delta = 0$ is also unforgiving of any small inconsistencies in labeling. Ignoring these limitations, E2E-Spot outperforms the baseline approaches, and compares similarly to models using hand-engineered pose features, in agreement with human annotators (Table C5). The practical meaning of mAP @ $\delta = 0$, however, is limited due to the aforementioned confounds.

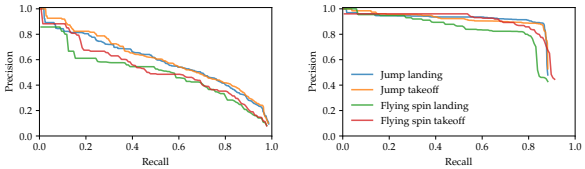
C.5 Visualizing the Spotting Performance of Different Classes

The difficulty of precisely spotting events can vary by event class. In Figure C1, we show interpolated precision-recall curves for the different classes in the Tennis, Figure Skating, FineDiving, and FineGym datasets from our default E2E-Spot 200MF model trained on RGB inputs.

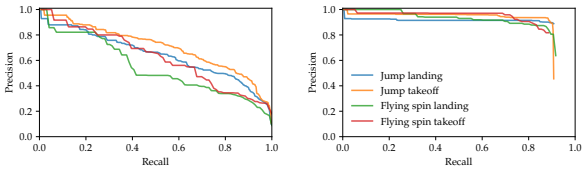
While spotting performance is similar among the different classes that comprise Tennis, Figure Skating, and FineDiving, spotting on FineGym shows a large amount of variation; some classes such as "balance beam dismounts start" and "floor exercise front_salto start" are spotted with high precision and recall at $\delta = 1$, while other classes such as "vault (timestamp 0)" and "balance beam turns end" exhibit much lower performance. We noted in § 4 that there is variation in the visual precision of different FineGym classes, where the annotated frames do not necessarily map to salient visual events.



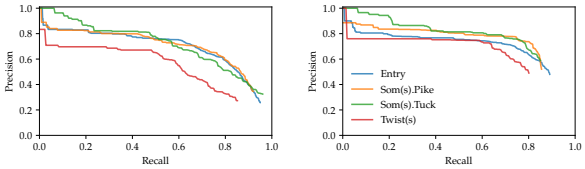
(a) Tennis (6 classes)



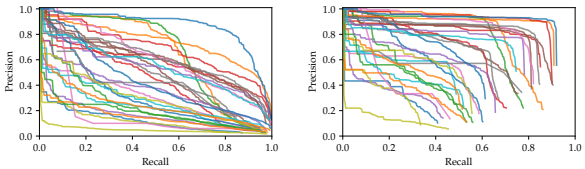
(b) FS-Comp (4 classes)



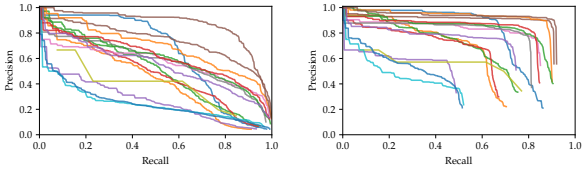
(c) FS-Perf (4 classes)



(d) FineDiving (4 classes)



(e) FineGym-Full (32 classes)



(f) FineGym-Start (16 classes)

Fig. C1: **Precision-recall curves for each event class at $\delta = 1$**, produced by E2E-Spot's default configuration. Charts on the **left are without NMS** and charts on the **right are with NMS**. NMS improves precision by suppressing nearby detections but can also lead to lower recall.

D Dataset Details

We use the Tennis [28], Figure Skating [13], FineDiving [25], and FineGym [20] datasets, with precise temporal event labels.

D.1 Tennis

The Tennis dataset is an extension of the dataset proposed by Zhang et al [28] to 19 new videos. Like the nine original videos from [28], the new videos are obtained from YouTube at full HD resolution and contain content from the US Open and Wimbledon tournaments. We annotate at least one full 'set' (a unit of gameplay) from each of the 19 new videos in order to diversify the dataset for training and evaluation.

To focus on the temporal aspect of precise spotting, we evaluate on the six top-level categories of events enumerated in § 4 and Table D6. These events are selected by their temporal definitions instead of the full set of semantic action attributes (e.g., swing type differentiated by topspin vs. slice; forehand vs. backhand; volley). The dataset contains 1.3M frames, of which 2.6% are precise temporal events.

D.2 Figure Skating

We extend the labels by Hong et al [13], which include fine-grained action classes and their temporal extents at approximately 1 second precision. To perform precise spotting, we manually re-annotate the labels to frame-accurate take-off and landing events.

As in Tennis, we separate temporally precise spotting from fine-grained classification of actions (e.g., the jump type) in order to focus on the temporal aspect of the spotting problem. See Table D7 for event statistics. The dataset contains 1.6M frames, of which only 0.23% are precise temporal events.

D.3 FineDiving

We use the pre-extracted frames provided by Xu et al [25] and spot the frames of transition between segments. The events include somersaults.pike, somersaults.tuck, twists, and entry. Note that we ignore the number of revolutions when generating frame-level event labels. See Table D8 for event statistics. The dataset contains 547K frames, of which 2.2% are precise temporal events.

D.4 FineGym

FineGym is a large gymnastics video dataset released by Shao et al [20]. It contains annotations for balance beam, floor exercises, uneven bars, and vaulting. The dataset is primarily used for action recognition, with 288 fine-grained classes

and their time intervals. These actions are contained within individual performances (e.g., an untrimmed balance beam routine), and several performances appear in a single video from YouTube.

We detect precise events within the untrimmed performances and split the dataset three ways for training, validation, and testing; these splits do not contain overlap in performances and source videos. We discard any performances that do not contain temporal annotations, have malformed annotations, or have annotations that are missing a class label in Gym288, leaving 5,374 performances.

Shao et al. [20] propose a hierarchy of action categories (to which the Gym288 classes belong), and we reduce the spotting problem to the granularity of these categories (e.g., "balance beam dismounts" is one example). Because our focus is temporal precision, we leave the challenging task of (unbalanced) 288-way action classification, which can be performed after events have been spotted, to past and future work on fine-grained action recognition.

We define temporally precise events in FineGym as the start and end frames of action intervals. This definition is straightforward for actions in balance beam, floor exercises, and uneven bars. Each vault, however, is specified as a sequence of three back-to-back segments, which we convert into four events. See Table D9 for the event breakdown and statistics.

A minority of videos (259) in the FineGym dataset have frame rates higher than 25–30 FPS. For consistency, since our spotting tolerances are defined in δ frames, we resample those videos to between 25–30 FPS. The final dataset contains 7.6M frames, 1.1% of which are precise temporal events.

Table D6: **Tennis dataset:** event classes and their counts.

Event class	Train	Val	Test
Near-court serve (ball contact)	673	238	779
Near-court swing (ball contact)	2199	709	4136
Near-court ball bounce	2606	871	4650
Far-court serve (ball contact)	657	200	800
Far-court swing (ball contact)	2220	757	4146
Far-court ball bounce	2621	867	4662

Table D7: **Figure Skating dataset:** event classes and their counts.

	FS-Comp			FS-Perf		
Event class	Train	Val	Test	Train	Val	Test
Jump takeoff	704	233	527	723	372	369
Jump landing	704	233	527	723	372	369
Flying spin takeoff	178	59	136	183	94	96
Flying spin landing	178	59	136	183	94	96

Table D8: **FineDiving dataset:** event classes and their counts.

Event class	Train	Val	Test
Entry	1794	449	741
Som(s).Pike	1254	345	553
Som(s).Tuck	667	149	255
Twist(s)	467	120	216

Table D9: **FineGym dataset:** event classes and their counts. The classes are based on the 'set-level categories' defined by Shao et al [20]. We refer to the full set of classes as FineGym-Full and a more visually consistent subset, containing primarily start events, as FineGym-Start.

Event class	In FineGym-Start	Train	Val	Test
Floor exercise leap_jump_hop start	✓	2007	602	629
Floor exercise leap_jump_hop end		2007	602	629
Floor exercise turns start	✓	683	197	223
Floor exercise turns end		683	197	223
Floor exercise side_salto start	✓	23	13	13
Floor exercise side_salto end		23	13	13
Floor exercise front_salto start	✓	818	259	268
Floor exercise front_salto end		818	259	268
Floor exercise back_salto start	✓	1850	524	604
Floor exercise back_salto end		1850	524	604
Balance beam leap_jump_hop start	✓	3062	765	960
Balance beam leap_jump_hop end		3062	765	960
Balance beam turns start	✓	857	215	299
Balance beam turns end		857	215	299
Balance beam flight_salto start	✓	2637	720	830
Balance beam flight_salto end		2637	720	830
Balance beam flight_handspring start	✓	1835	440	618
Balance beam flight_handspring end		1835	440	618
Balance beam dismounts start	✓	763	188	267
Balance beam dismounts end		763	188	267
Uneven bars circles start	✓	4143	1151	1318
Uneven bars circles end		4143	1151	1318
Uneven bars flight_same_bar start	✓	1029	270	325
Uneven bars flight_same_bar end		1029	270	325
Uneven bars transition_flight start	✓	2079	630	680
Uneven bars transition_flight end		2079	630	680
Uneven bars dismounts start	✓	750	225	252
Uneven bars dismounts end		750	225	252
Vault (timestamp 0)		1263	367	401
Vault (timestamp 1)	✓	1263	367	401
Vault (timestamp 2)	✓	1263	367	401
Vault (timestamp 3)		1263	367	401

References

1. Alwassel, H., Giancola, S., Ghanem, B.: TSP: Temporally-sensitive pretraining of video encoders for localization tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 3173–3183 (October 2021)
2. 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 (CVPR) (July 2017)
3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: Proceedings of NIPS Deep Learning and Representation Learning Workshop (2014)
4. Cioppa, A., Deliege, A., Giancola, S., Ghanem, B., Van Droogenbroeck, M., Gade, R., Moeslund, T.B.: A context-aware loss function for action spotting in soccer videos. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
5. 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 (CVPR) Workshops. pp. 4508–4519 (June 2021)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2009)
7. Fan, H., Murrell, T., Wang, H., Alwala, K.V., Li, Y., Li, Y., Xiong, B., Ravi, N., Li, M., Yang, H., Malik, J., Girshick, R., Feiszli, M., Adcock, A., Lo, W.Y., Feichtenhofer, C.: PyTorchVideo: A deep learning library for video understanding. In: Proceedings of the 29th ACM International Conference on Multimedia (2021)
8. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6824–6835 (October 2021)
9. Farha, Y.A., Gall, J.: MS-TCN: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
10. Giancola, S., Ghanem, B.: Temporally-aware feature pooling for action spotting in soccer broadcasts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 4490–4499 (June 2021)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
13. 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 (ICCV). pp. 9254–9263 (October 2021)
14. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The Kinetics human action video dataset (2017), arXiv:1705.06950

15. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
16. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019)
17. Nvidia: Nvidia RTX A5000 data sheet (2021)
18. PyTorch: Pytorch documentation: Transformer (2022)
19. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollar, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
20. 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 (CVPR) (June 2020)
21. 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 (CVPR) (June 2020)
22. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: Proceedings of the European Conference on Computer Vision (ECCV) (August 2020)
23. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
24. Wightman, R.: PyTorch image models. `https://github.com/rwightman/pytorch-image-models` (2019)
25. 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 (CVPR). pp. 2949–2958 (June 2022)
26. 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 (CVPR) (June 2020)
27. Yi, F., Wen, H., Jiang, T.: ASFormer: Transformer for action segmentation. In: Proceedings of the British Machine Vision Conference (BMVC) (November 2021)
28. Zhang, H., Sciutto, C., Agrawala, M., Fatahalian, K.: Vid2Player: Controllable video sprites that behave and appear like professional tennis players. ACM Transactions on Graphics **40**(3) (2021)
29. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
30. Zhou, X., Kang, L., Cheng, Z., He, B., Xin, J.: Feature combination meets attention: Baidu soccer embeddings and transformer based temporal detection (2021), arXiv:2106.14447