# A Neuro-Symbolic Approach for Real-World Event Recognition from Weak Supervision

**Gianluca Apriceno**
University of Trento, Italy
Fondazione Bruno Kessler, Trento, Italy

**Andrea Passerini**
University of Trento, Italy

**Luciano Serafini**
Fondazione Bruno Kessler, Trento, Italy

## Abstract

Events are structured entities involving different components (e.g, the participants, their roles etc.) and their relations. Structured events are typically defined in terms of (a subset of) simpler, atomic events and a set of temporal relation between them. Temporal Event Detection (TED) is the task of detecting structured and atomic events within data streams, most often text or video sequences, and has numerous applications, from video surveillance to sports analytics. Existing deep learning approaches solve TED task by implicitly learning the temporal correlations among events from data. As consequence, these approaches often fail in ensuring a consistent prediction in terms of the relationship between structured and atomic events. On the other hand, neuro-symbolic approaches have shown their capability to constrain the output of the neural networks to be consistent with respect to the background knowledge of the domain. In this paper, we propose a neuro-symbolic approach for TED in a real world scenario involving sports activities. We show how by incorporating simple knowledge involving the relative order of atomic events and constraints on their duration, the approach substantially outperforms a fully neural solution in terms of recognition accuracy, when little or even no supervision is available on the atomic events.

**2012 ACM Subject Classification** Computing methodologies → Temporal reasoning; Computing methodologies → Activity recognition and understanding

**Keywords and phrases** structured events, temporal event detection, neuro-symbolic integration

**Digital Object Identifier** 10.4230/LIPIcs.TIME.2022.12

**Supplementary Material** *Software (Source Code)*: `https://github.com/Gianlu94/A-neuro-symbolic-approach-for-real-world-event-recognition-from-weak-supervision`

**Funding** This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

## 1 Introduction

Events are structured entities that involve multiple components, like the participants, their role, the type and the atomic events composing it. For example, in athletics, the event *high jump* involves one person (the athlete), performing the atomic events *run*, *jump* and *fall* in sequence. One of the main challenging tasks is temporal event detection (TED) that consists in detecting events within data stream, like text and video. Continuing the example, it consists in identifying the class of the atomic events and the interval of time where they occurred. Many sub-symbolic approaches, mostly based on neural networks, have been proposed for event recognition [1, 25]. One of the main drawbacks of these kind of approaches is the amount of training data. Indeed, having a large training set is fundamental for an appropriate and effective training of the model. Furthermore, "rich" annotations at different levels are required in order to solve the task (e.g., frame-by-frame annotation of atomic events). Large amounts of deeply annotated data are hard to collect. Additionally, errors


© Gianluca Apriceno, Andrea Passerini, and Luciano Serafini;
licensed under Creative Commons License CC-BY 4.0

in the annotations (e.g. in case of crowdsourced ones) may introduce noise in the model and compromise its accuracy. More importantly, purely neural approaches cannot guarantee consistency of the predictions with the domain knowledge, in terms of the relationship between the structured event and the atomic events that compose it. Neuro-symbolic approaches [11] have recently gained increasing popularity as a means to make the best of both worlds, by combining the effectiveness in low-level processing of deep learning technology with the ability of symbolic approaches to express complex domain knowledge. Popular frameworks including DeepProbLog [16], DeepStochLog [24], Logic Tensor Networks [6], LYRICS [17] and NeurASP [28] have been proposed and applied to solve different structured tasks, like Semantic Image Interpretation [8]. In the context of event recognition, the DeepProbLog framework has proved effective in recognizing both structured and simple events as well as generalizing to unseen outcomes [3, 23]. However, these results have been obtained on artificial scenarios, and the framework has serious issues of scalability when the complexity of the setting increases [9].

In this paper, we present a neuro-symbolic approach for structured event recognition in sport videos. The task is out of reach of popular neuro-symbolic frameworks like DeepProbLog, because of the computational complexity given by number of frames in a video combined with the temporal constraints of the background knowledge. We tackle the problem by combining neural predictions on individual frames with a mixed integer linear programming formulation enforcing satisfaction of (soft) temporal constraints from the background knowledge and similarity with the neural outputs. The approach is fully differentiable and end-to-end trainable.

Our experimental evaluation shows how the neuro-symbolic approach provides substantially more accurate predictions with respect to a fully neural solution, with the additional feature of guaranteeing that predictions satisfy existing hard constraints. Quite remarkably, the approach is capable of predicting the sequence of atomic events that constitute a structured event even without having any supervision on them, by simply leveraging background knowledge in terms of duration constraints to guide the learning of the underlying neural network.

The rest of the paper is structured as follows: Section 2 briefly reviews the state of the art approaches that have been proposed for event recognition; Section 3 formally defines the problem; Section 4 describes our proposed approach; Section 5 presents the experimental setting; Section 6 reports the experimental results. Finally, Section 7 draws some concluding remarks and discusses directions for future work.

## 2 State of the art

Event recognition has always attracted researchers coming from different fields, like Computer Vision and NLP. The particular attention towards event recognition may be motivated by its multiple data stream nature and by its impact in people's daily life. Looking at the literature, approaches to event recognition can be classified into three categories: sub-symbolic, symbolic and neuro-symbolic. Sub-symbolic approaches (mostly based on neural networks) moved from manually crafted features to automatic features learning (see [1] and [25] for a survey). These approaches require a huge amount of training data with a rich annotation at different levels (e.g, the type and temporal location of the event). These data are hard to collect and it is difficult to ensure high-quality annotations for large amounts of example, so that high levels of annotation errors can affect the accuracy of the networks being trained. Furthermore, the trained model is a black box model that cannot explain its decisions and is not guaranteed

to be consistent with existing background knowledge. An attempt to make sub-symbolic approaches more interpretable is represented by Concept Bottleneck Models [15] where the activation of (a subset of) human-specified concepts is used to explain the model's final decision. However, works like [15] focus on atemporal domains (e.g. image). In [12], authors propose a novel approach that addresses concepts explanation in videos, but they are not able to capture spatial and temporal relationships between concepts. On the other hand, symbolic approaches like [5], are explainable and knowledge-consistent, but are not robust in the presence of noise. Therefore, symbolic approaches dealing with uncertainty have been proposed [2]. In [20, 4], authors recognize higher events by combining evidence of simple events with domain knowledge using the probabilistic logic programming framework ProbLog [18] . In this case, knowledge on low level events is assumed to be given. Recently, neuro-symbolic approaches have started to be applied in the context of event recognition. In works like [13, 14, 27, 22, 10], pre-trained neural networks are used to extract lower events and then passed to a symbolic layer that encodes the knowledge of the domain in the form of logic rules. In [26], authors propose an end-to-end model where a neural network is also used to learn to simulate the symbolic layer. One of the drawbacks is that the neural network has to be re-trained in case of even minimal changes/updates of the existing knowledge. The DeepProbLog neuro-symbolic framework [16] has been employed in a couple of recent studies [3, 23] to perform structured event recognition in videos and audio streams respectively. By using DeepProbLog, any change/update to knowledge can be easily integrated. Both [3] and [23] perform event recognition in artificial scenarios. This may be motivated by the scalability issue of DeepProbLog, that is particularly acute when considering tasks involving time. Indeed, the authors of the NESTER framework [9] showed how an optimization modulo theory [19] reasoning layer refining neural network predictions is substantially more efficient than a solution based on DeepProbLog on a toy handwritten equation recognition task. The approach however assumes complete supervision on the neural network outputs at training time, and does not address temporal event detection. Our solution adapts this idea to address structured and atomic event recognition on real-world video streams. In this case, the reasoning layer is also used to provide supervision when limited/no labels for the events are available.

## 3 Problem Definition

Our problem can be summarized as follows: Given a data sequence $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{l}$ of real-value tensors $\boldsymbol{x}_i$ and some background knowledge $\mathcal{K}$ about the relationship between structured and atomic events, we are interested in providing a description of the atomic and structured events that are happening during the sequence. Let us now specify all the details of the problem. To represent background knowledge about structured and atomic events we use a variation of the event calculus based on First Order Logic. Let $\mathcal{L}$ be a first order language that contains a set of constants $\mathcal{E}$ for event types, which is partitioned in two disjoint sets of constants $\mathcal{A}$ and $\mathcal{S}$ for atomic and structured events, respectively. $\mathcal{L}$ contains also the set of constants $\mathbb{N}$ of natural numbers which are used to denote time points. The set of predicates of $\mathcal{L}$ includes the equality predicate, the order relation defined as usual on time points, and the ternary predicate $happens(e, t_1, t_2)$, where $e$ is a term for an event type and $t_1$, and $t_2$ are terms for time points. The atom $happens(e, t_1, t_2)$ represents the proposition that an event of type $e$ starts at time $t_1$ and terminates at time $t_2$ (not included). In addition to the usual axioms for the equality and order relation we have the axiom

$$\forall xyz(happens(x, y, z) \rightarrow y < z)$$

As far as the semantics of $\mathcal{L}$, we consider the set of Herbrand Interpretations of $\mathcal{L}$, i.e., all the subsets of the Herbrand Base $\mathcal{H}$ defined as

$$\mathcal{H} = \{happens(e, t_1, t_2) \mid e \in \mathcal{E}, t_1 < t_2 \in \mathbb{N}\}$$

Now we are ready to provide a more precise formulation of our problem: Given a knowledge base $\mathcal{K}$ in the language $\mathcal{L}$ that expresses general knowledge about the event types in $\mathcal{E}$, and a data sequence $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{l}$, we have to find an herbrand interpretation $\mathcal{I}$, such that $\mathcal{I} \models \mathcal{K}$, and such that $happens(e, t_1, t_2) \in \mathcal{I}$ if and only if the data sub-sequence $\boldsymbol{X}_{t_1:t_2} = \{\boldsymbol{x}_i\}_{i=t_1}^{t_2-1}$ shows that an event of type $e$ is happening. Intuitively, $\mathcal{I}$ describes the type and the class of the events happening in $\boldsymbol{X}$ and when they happened.

▶ **Example 1.** Let $\boldsymbol{X}$ be a video where a person is doing a high jump (structured event) in the interval $[1, 31]$. The background knowledge contains the fact that a high jump can be decomposed into a sequence of three atomic events: run, jump and fall, which can be expressed by the following formula:

$$\begin{aligned}\forall b_{hj} e_{ij} (happens(highjump, b_{hj}, e_{hj}) \leftrightarrow \exists\, b_r, e_r, b_j, e_j, b_f, e_f (\\ happens(run, b_r, e_r) \wedge happens(jump, b_j, e_j) \wedge happens(fall, b_f, e_f) \wedge \\ b_r = b_{hj} \wedge e_r = b_j \wedge e_j = b_f \wedge e_f = e_{hj}))\end{aligned}$$

Two examples of interpretations that satisfy the above constraints are:

$$\begin{aligned}\mathcal{I}_1 = \{&happens(highjump, 1, 31), happens(run, 1, 21), happens(jump, 21, 25),\\ &happens(fall, 25, 31)\}\\ \mathcal{I}_2 = \{&happens(highjump, 1, 31), happens(run, 1, 23), happens(jump, 23, 28),\\ &happens(fall, 28, 31)\}\\ &\vdots\end{aligned}$$

Since we may have more than one interpretation that satisfy $\mathcal{K}$, we define a cost function $c : I \to R$ and select the interpretation $I_c^*$ with the minimum cost:

$$I_c^* = \underset{I_c \models \mathcal{K}}{\operatorname{argmin}}\, c(I_c)$$

In order to find $I_c^*$, we define a neuro-symbolic approach that combines low-level neural processing with high level reasoning in terms of background knowledge on the events.
The kind of supervision we provide to train a neuro-symbolic model in order to find $I_c^*$ is:

$$\left\{\boldsymbol{X}^{(i)}, G_a^{(i)}\right\}_{i=1}^{n}$$

where $G_a^{(i)}$ is a set of ground atoms which are true in $\boldsymbol{X}^{(i)}$. Supervision is always assumed to be partial, including the case in which supervision is limited to structured events, and atomic events need to be learned in a fully unsupervised way. See experiments for the details.

## 4 Proposed Approach

Our objective consists in finding an interpretation $I$ that has to explain what happened in $\boldsymbol{X}$ both in terms of structured and atomic events. In order to achieve it, we have to recognize the classes of structured and atomic events happening in $\boldsymbol{X}$ and the (interval of)

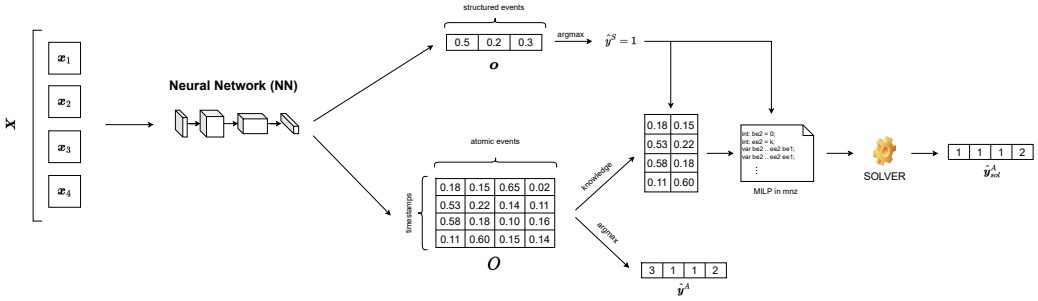



**Figure 1** Inference steps of our neuro-symbolic approach on a data sequence of length 4, with 3 structured events and 4 atomic events. The structured event of class 1, which is the one predicted by the NN, is defined in terms of the sequence of atomic events 1 and 2, respectively.

time where they occurred. To achieve this objective, we use an end-to-end differentiable neuro-symbolic approach that combines low level processing of a neural network with a logic layer that leverages knowledge about structured and atomic events. An overall overview of our neuro-symbolic approach is depicted in Figure 1. As can be seen by looking at the figure, the first step consists in passing $\boldsymbol{X}$ to a neural network NN. NN may be any kind of network (e.g., Convolutional, RNN and LSTM) and has two different heads, one for structured and one for atomic events. The head for the structured events returns as output a vector $\boldsymbol{o}$ where $o_i \in [0,1]$, with $i = 1, \ldots, k$ (assuming $k$ is the number of structured events), is the probability of the $i-th$ structured event. On the other hand, the head for the atomic events returns a matrix $O \in [0,1]^{l \times n}$ where entry $O[i,j]$, with $i = 1, \ldots, l$ and $j = 1, \ldots, n$ (assuming $l$ and $n$ are the length of $\boldsymbol{X}$ and the number of atomic events, respectively), represents the probability that event $j$ happens at timestamp $i$. The predicted structured event for a video $\boldsymbol{X}$ is the one maximizing the probability of the corresponding output head of NN, i.e.:

$$\hat{y}^{\mathcal{S}} = \underset{i=1,\ldots,k}{\text{argmax}}\ o_i$$

In principle, the sequence of atomic events could be predicted in a similar fashion by maximizing for each frame the probability of the atomic event head of NN at that frame, i.e.:

$$\hat{y}_i^{\mathcal{A}} = \underset{1 \leq j \leq l}{\text{argmax}}\ O[i,j] \tag{1}$$

Indeed, this is how our fully-neural baseline works. However, the vector $\hat{\boldsymbol{y}}^{\mathcal{A}}$ of atomic event predictions for the frames of a video $\boldsymbol{X}$ may contain inconsistencies (e.g., predicting the atomic event *jump* as part of a *javelinthrow* structured event, predicting *fall* before *jump* within a *highjump*, or even predicting a *jump* that is much longer than the *run* that precedes it). Our neuro-symbolic architecture prevents these inconsistencies by combining neural network predictions with hard and soft constraints provided by the domain knowledge. The domain knowledge we exploit is quite simple, and provides hard constraints determining the sequence of atomic events that constitute a structured event, and soft constraints about minimal and maximal duration of each atomic event and relative duration between atomic events making up a structured event. Table 1 reports an example of the hard constraints that we consider for the *javelinthrow* structured event. Similar constraints are generated for the other structured events. Given the structured event $\hat{y}^{\mathcal{S}}$ predicted by NN, the corresponding sequence of atomic events is computed by solving a MILP problem encoding the (hard

**Table 1** Example of hard constraints for the *javelinthrow* structured event, divided into generic constraints that hold for any structured event, and those specific of the *javelinthrow* event.

| Hard Constraints | |
|---|---|
| Generic Constraints (assuming $k$ atomic events) | |
| $e_i > b_i \quad \forall\, i$ | Events should end after they began |
| $b_1 = 1 \wedge e_k = l$ | Sequence of atomic events should span the whole clip |
| $e_i = b_{i+1} - 1 \quad \forall\, i \in 0 \dots l-1$ | No gap among consecutive events |
| Specific Constraints (for the *javelinthrow* structured event) | |
| $a_1 = run \wedge a_2 = throw$ | *javelinthrow* is a *run* followed by a *throw* |
| $d_1 > d_2$ | *run* should take longer than *throw* |

and soft) constraints combined with a scoring function measuring the compatibility of the sequence of atomic events with the NN outputs $O$. The MILP problem for a structured event (we have a separate problem for each possible structured event) is defined as follows:

$$\begin{aligned} &\underset{V}{\text{minimize}} \quad -f(V,O) + \sum_{j=1}^{m_s} \xi_t c_j(V) \\ &\text{subject to} \quad h_i(V) \quad \forall\, i = 1, \dots, m_h \end{aligned} \tag{2}$$

Here $V$ is a sequence of triplets $(a, b, e)$, where $a \in \mathcal{A}$ is an atomic event, $b, e \in \mathbb{N}$ are the starting and ending frames of the event respectively. The number of atomic events is determined by the structured event being modelled. The scoring function $f(V, O)$ computes the compatibility of $V$ with $O$ as follows:

$$f(V,O) = \sum_{(a,b,e) \in V} \left( \sum_{i=b}^{e} O[i,a] - \sum_{j=1}^{b-1} O[j,a] - \sum_{j=e+1}^{l} O[j,a] \right)$$

It basically computes the sum of the probabilities of each atomic event in the range in which it is predicted, and subtracts its probability outside of this range ($l$ is the overall length of the video clip).

The soft constraints $c_j(V)$ encode duration ranges for atomic events or combination of atomic events. For instance, the constraint that the sum of the durations of *run* and *jump* should be within the sum of the maximal and minimal durations respectively is formalized as follows:

$$\min(|d_1 + d_2 - max_{run} - max_{jump}|, |d_1 + d_2 - min_{run} - min_{jump}|)$$

where:

$$d_1 = e_1 - b_1 + 1,\ d_2 = e_2 - b_2 + 1$$

$$a_1 = run,\ a_2 = jump$$

The hard constraints $h_i(V)$ encode temporal relations between atomic events and with respect to the structured event. See Table 1 for examples. Intuitively, the solutions of the MILP problem for the predicted structured event $\hat{y}^S$ where only hard contraints $h_i(V)$ are

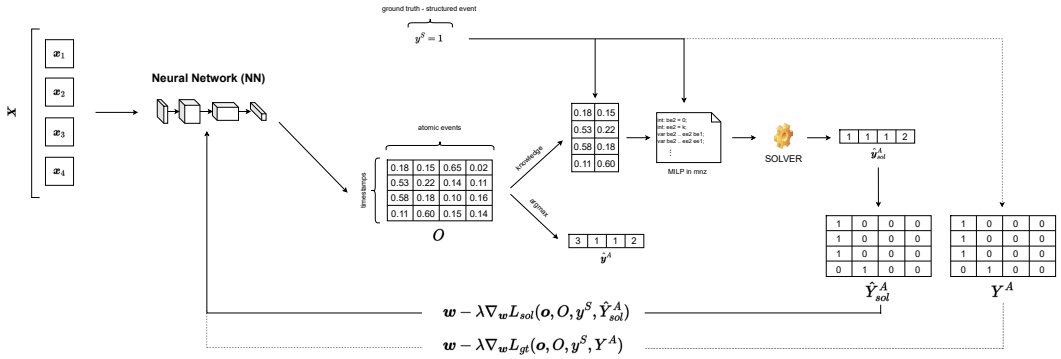



**Figure 2** Training of our neuro-symbolic approach.

considered, provide a set of candidate interpretations for $\boldsymbol{X}$. By including the objective $f(V, O)$ and the soft constraints $c_j(V)$ we obtain the interpretation with the minimum cost for $\boldsymbol{X}$ (i.e, $Y_c^*$). The label vector $\hat{y}^{\mathcal{A}}_{sol}$ in Figure 1, which is the neuro-symbolic counterpart of $\hat{y}^{\mathcal{A}}$ in Eq. 1, is obtained by "unrolling" the optimal $V$ into an atomic label for each frame in its predicted range.

▶ **Example 2.** Let $\boldsymbol{X}$ a video of length 20 where a person is performing a structured event, but we do not know which kind of structured event. Now, suppose we have, in addition to the structured event highjump of example 1, the structured event javelin throw and that the background knowledge contains the fact that a javelin throw can be decomposed into a sequence of two atomic events: run and jump, which can be expressed by the following formula:

$$\forall b_{jt} e_{jt} (happens(javelinthrow, b_{jt}, e_{jt}) \leftrightarrow \exists\, b_r, e_r, b_t, e_t, ( \\ happens(run, b_r, e_r) \wedge happens(throw, b_t, e_t) \wedge \\ b_r = b_{jt} \wedge e_r = b_t \wedge e_t = e_{hj}))$$

Also, suppose that the head of NN for structure events predicted $\hat{y}^{\mathcal{S}} = javelinthrow$ for $\boldsymbol{X}$. If we solve the MILP for the javelin throw where only hard constraints are considered (for example the ones in table 1), we have that some of the candidate interpretations will be:

$$\mathcal{I}_1 = \{happens(javelinthrow, 1, 21), happens(run, 1, 13), happens(throw, 13, 21)\}$$
$$\mathcal{I}_2 = \{happens(javelinthrow, 1, 21), happens(run, 1, 17), happens(throw, 17, 21)\}$$
$$\vdots$$

By considering and solving the whole MILP, we obtain $I_c^*$. Continuing the example, if we have a soft constraint which penalizes interpretations having short duration for *run*, we have that $I_c^* = I_2$, that corresponds to the solution $V = [(run, 1, 17), (jump, 17, 21)]$ of Problem 2.

Figure 2 shows the training process of the architecture. As we assume that the ground-truth $y^{\mathcal{S}}$ of $\boldsymbol{X}$ is available at training time, the head for the structured events is not used anymore to predict $\hat{y}^{\mathcal{S}}$, but the ground-truth itself is used instead. Furthermore, if the ground-truth for the atomic events ($Y^{\mathcal{A}} \in R^{l \times n}$) is also available, we use it to train the head of the atomic events. If this information is not available, we use pseudo-labels generated from the architecture. The generation of such pseudo-labels consists of an inference step

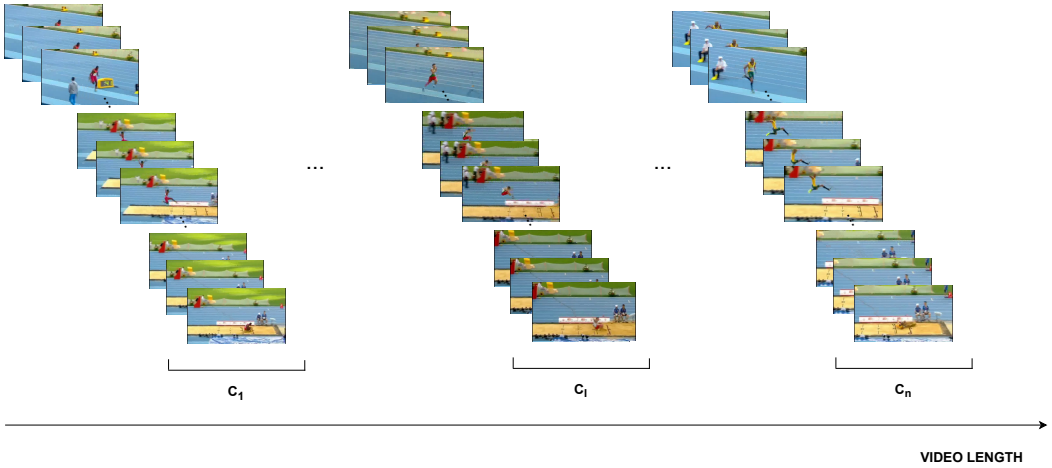



**Figure 3** Extraction of clips of structured events from an untrimmed video.

in the currently trained architecture as per Figure 1, with the only difference that the structured event is given by the ground-truth $y^{\mathcal{S}}$ rather than the NN output. The atomic event prediction vector $\hat{\boldsymbol{y}}^{\mathcal{A}}_{sol}$ is turned into a binary label matrix $\hat{Y}^{\mathcal{A}}_{sol} \in \{0,1\}^{l \times n}$ by one-hot encoding atomic labels (i.e., $\hat{Y}^{\mathcal{A}}_{sol}[i,j]$ is set to 1 if $j = \hat{y}^{\mathcal{A}}_{sol}[i]$ and 0 otherwise). Then, we define two losses:

$$L_{gt}(\boldsymbol{o}, O, y^{\mathcal{S}}, Y^{\mathcal{A}}) = L(\boldsymbol{o}, y^{\mathcal{S}}) + L(O, Y^{\mathcal{A}}) \tag{3}$$

$$L_{sol}(\boldsymbol{o}, O, y^{\mathcal{S}}, \hat{Y}^{\mathcal{A}}_{sol}) = L(\boldsymbol{o}, y^{\mathcal{S}}) + L(O, \hat{Y}^{\mathcal{A}}_{sol}) \tag{4}$$

Where Loss 3 refers to the case where both ground-truth (structured and atomic) are available, while Loss 4 refers to the case where the ground-truth for structured events is available and the ground-truth for the atomic events is not, and, then, we use the pseudo labels. In order to train NN to recognize both kind of the events, we minimize, depending on the case, one of the aforementioned loss and use gradient descent to update its weights.

## 5 Experimental setting

Our experimental setting has the aim to show if our proposed neuro-symbolic approach leads to an advantage in the recognition of both structured and atomic events with respect to a fully neural approach, when both approaches are trained with weak and limited amount of supervision in terms of events. In details, we want to see how the knowledge is able to compensate in the case when no or few and potentially noisy labels for events are available. To achieve this objective, we first need a dataset of structured and atomic events. We build such dataset from the Multi-THUMOS untrimmed video dataset [29]. Since in [29], there is no explicit distinction between structured and atomic events, we define it and splits the events according. In particular, we consider as structured events those events that can be decomposed as a sequence of other (atomic) events, and cut each video into clips corresponding to structured events (Figure 3).

For each structured event, we do not consider all the clips, but we remove those clips where some of the atomic events defining the structured event are not present (e.g, replay). The structured and atomic events we consider are shown in appendix B. Then, after building the dataset, we define the scenario. The scenario consists of clips of different length where,

in each clip, a person is performing one (and only one) structured event among the ones reported in appendix B. The learning setting we consider to evaluate the fully neural approach and our proposed neuro-symbolic approach consists in having full supervision at level of structured events and limited and potentially noisy (e.g., overlapping between atomic events) supervision in terms of atomic events. The kind of supervision we provide is as follows:

$$\{happens(highjump, 1, 50),\ happens(run, 1, 31),\ happens(jump, 31, 45),\ happens(fall, 45, 50)\}$$
$$\{happens(hammerthrow, 1, 30),\ happens(windup, 1, 15),\ happens(spin, 10, 25),\ happens(release, 25, 30)\}$$
$$\{happens(javelinthrow, 1, 30)\}$$

The first video is an example of noiseless labeling with full supervision on both structured and atomic events. The second video is fully supervised too, but atomic supervision is noisy, as there is an overlap between the *windup* and *spin* atomic events (this type of overlapping labeling is not rare in the dataset). The third video is a case where supervision is only provided at the structured event level, and there is no supervision on atomic events.

In this setting, we are interested in observing how the prediction in terms of atomic and structured events change when increasing the availability of data for atomic events. Furthermore, in the case of the neuro-symbolic approach, we are interested in seeing how complementing the supervision coming from the dataset with the supervision coming from the knowledge affects the predictions of the overall model. A noteworthy case is the one where no direct supervision at level of atomic events is provided at all, and then the model is completely trained with the supervision coming from the knowledge. The underlying model we use for both approaches is the one described in [21] where features extracted from a pre-trained two-stream I3D [7] are given as input in order to predict a matrix of events (more details about the model can be found in Appendix C). Differently from [21], we distinguish between structured and atomic events and consider two separate heads, as discussed in the previous section. About the training, we train the model for 20 epochs with learning rate of 1e−3 and weight decay of 1e−6, using Adam as optimizer. We also created a validation set of 10% of the training data in order to select the best model.

## 6 Results

In this section, we show and compare the results of the fully neural approach with respect to our proposed neuro-symbolic approach on the task described in Section 5. Figure 4 reports average $F_1$ scores of structured event prediction over 5 runs, for an increasing amount of supervision in terms of atomic events (from 0 to 100%). Note that in all cases supervision in terms of structured events is always provided. The green curve indicates the fully neural baseline, while the red curve indicates our neuro-symbolic approach. Results indicate that unsurprisingly, when there is full supervision on the structured event the addition of knowledge does not help in its identification.

Figure 5 reports average $F_1$ scores for the prediction of atomic events, again for a growing amount of supervision at the atomic level. The difference between the fully neural and the neuro-symbolic approach is striking. Substantial improvements of the neuro-symbolic approach can be observed for almost all atomic events. Only for the atomic events *run* and *jump*, we can see that the fully neural approach is really close to our approach. This is probably due to the temporal duration of these events, that is substantially higher than that

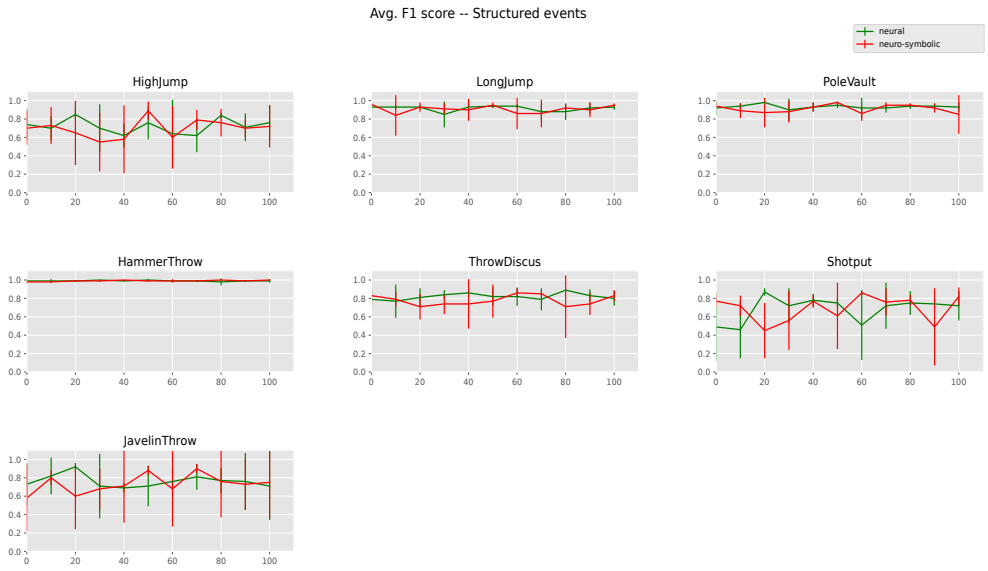



**Figure 4** $F_1$ scores on structured events averaged over 5 runs for the fully neural (green) and the neuro-symbolic (red) approaches, for increasing amount of supervision on atomic events.

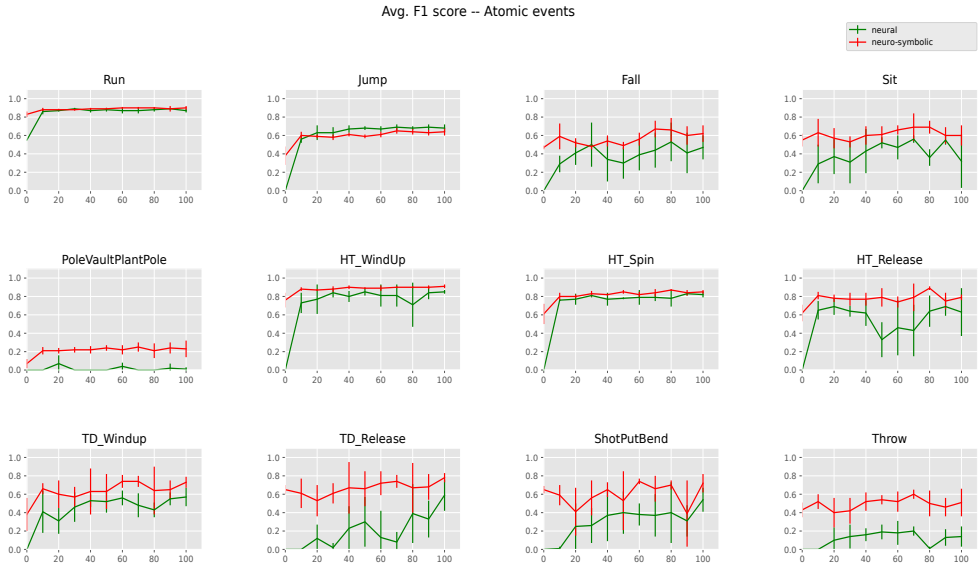



**Figure 5** $F_1$ scores on atomic events averaged over 5 runs for the fully neural (green) and the neuro-symbolic (red) approaches, for increasing amount of supervision on atomic events.

of the others. This implies that a reasonable number of frames labelled as *run* and *jump* will be available for the neural network even with a small fraction of labelled videos. On the other hand, atomic events like *release* and *throw* have a performance improvement that goes up as 60% and 50% respectively.

As stated in Section 5, a particularly significant case is the one where no direct supervision at all is provided at the level of atomic events. This corresponds to the leftmost point in the figures. The $F_1$ score of the fully neural approach is close to zero for almost all atomic events, corresponding to random guessing. On the other hand, the $F_1$ scores of the neuro-symbolic approach are often comparable to those of a fully supervised setting, showing how knowledge can be exploited to completely bypass the need for human supervision at the frame level, with major implications in terms of applicability and training costs.

Figure 6 shows some representative examples of the labeling provided by the two approaches highlighting the differences in prediction consistency between the two, both in terms of atomic events being detected and relative duration. Note that results are achieved with 100% supervision on atomic events, and highlight the importance of knowledge in guaranteeing the consistency of predictions. The Figure shows prediction for all frames in a video for three videos, a *highjump*, a *hammerthrow* and a *longjump* respectively. For each video, we compare ground truth atomic labels (yellow) with fully neural predictions (green) and neuro-symbolic ones (green). In the *highjump* case (top), the fully neural approach completely misses the last event and mispredicts it as part of the *jump* event. On the other hand, the neuro-symbolic approach correctly detects the last event a *fall*, and has a better estimate of the duration of each event. In the *hammerthrow* case (middle), the fully neural approach detects a *run* event, that cannot be part of a *hammerthrow*, and misses the *release* event. Again, the neuro-symbolic approach provides quite accurate estimates of the duration of each event, despite the short duration of *release* with respect to *windup* and *spin*. Finally, in the *longjump* case (bottom), the neural approach correctly identifies the initial *run*, but breaks the rest of the video into a sequence of short *jump*, *fall*, *sit* and even *run* events which is completely inconsistent, while the neuro-symbolic approach again accurately recovers both sequence and duration of the atomic events.

## 7 Conclusion and Future Work

In this work, we have proposed a neuro-symbolic approach for (structured and atomic) event recognition where knowledge about the events and their temporal relations is exploited both at training and inference time. We have instantiated our approach on a real-world scenario consisting of clips of sports events. Our experimental evaluation showed how our neuro-symbolic solution achieves substantial improvements over a fully neural baseline in terms of recognition of the atomic events that constitute a structured event. The approach is capable of learning to detect atomic events even with no supervision at all on them during training, by simply combining supervision on structured events, low-level neural processing and knowledge. While these results are promising, there are several directions which are left open for future research. A major direction consists in increasing the complexity of the scenarios being considered, by dealing with structured events involving multiple actors and complex relationships between the events, without making the underlying reasoning problem prohibitively expensive, something other neuro-symbolic frameworks currently struggle with.

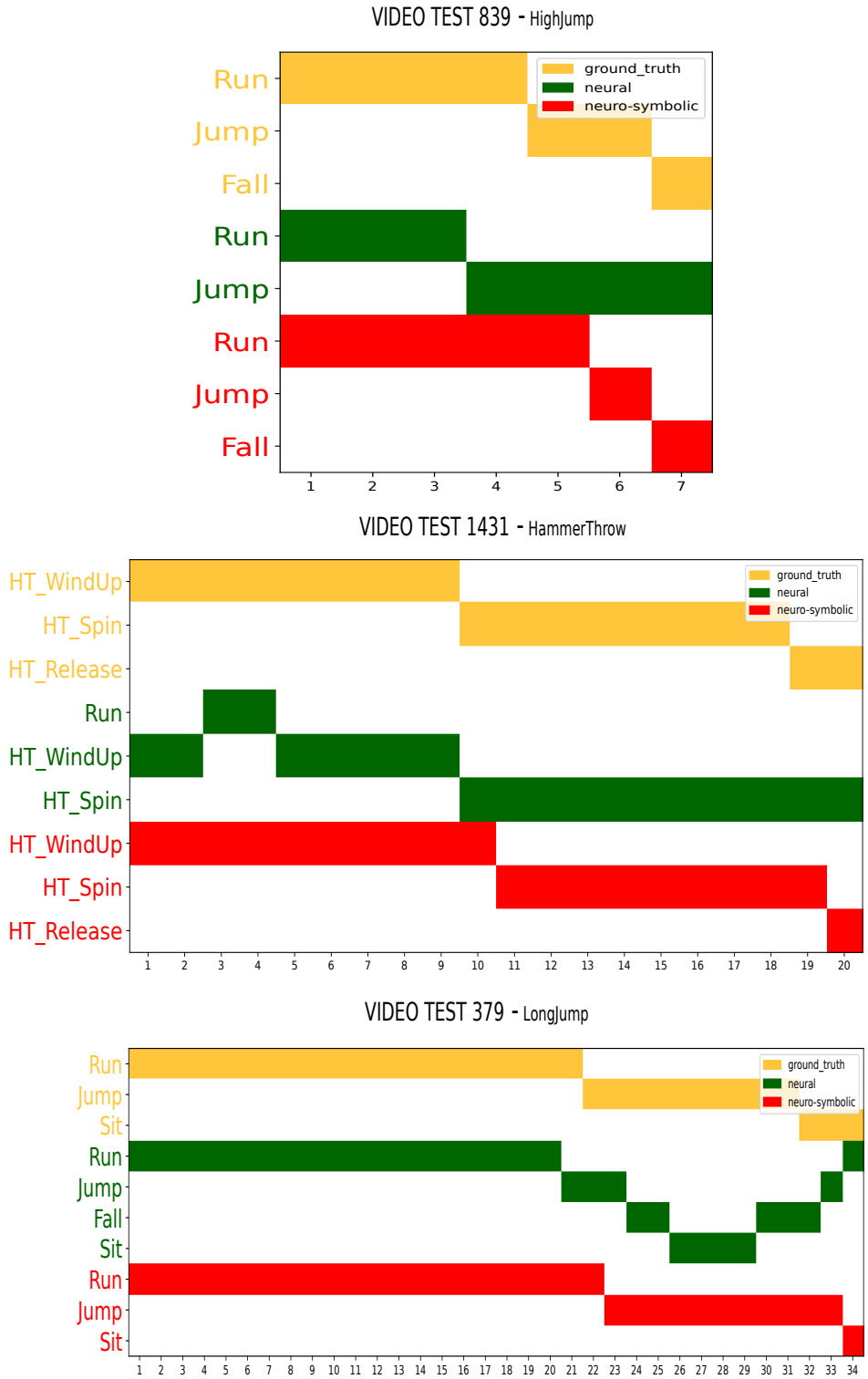



**Figure 6** Prediction of the sequence of atomic events for three clips representing a *highjump* (top), a *hammerthrow* (middle) and a *longjump* (bottom) respectively. Ground truth is in yellow, while the neural and neuro-symbolic predictions are in green and red respectively. Both models were trained with 100 % supervision on the atomic events. Clips were selected to show examples of inconsistencies in neural predictions.

## References


1 Kashif Ahmad Ahmad and Nicola Conci. How Deep Features Have Improved Event Recognition in Multimedia: A Survey. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 15(2):39:1–39:27, 2019. doi:10.1145/3306240.

2 Elias Alevizos, Anastasios Skarlatidis, Alexander Artikis, and Georgios Paliouras. Probabilistic Complex Event Recognition: A Survey. *ACM Computing Surveys*, 50(5):71:1–71:31, 2017. doi:10.1145/3117809.

3 Gianluca Apriceno, Andrea Passerini, and Luciano Serafini. A neuro-symbolic approach to structured event recognition. In Carlo Combi, Johann Eder, and Mark Reynolds, editors, *28th International Symposium on Temporal Representation and Reasoning, TIME 2021, September 27-29, 2021, Klagenfurt, Austria*, volume 206 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.TIME.2021.11.

4 Alexander Artikis, Evangelos Makris, and Georgios Paliouras. A probabilistic interval-based event calculus for activity recognition. *Annals of Mathematics and Artificial Intelligence*, 89(1-2):29–52, 2021. doi:10.1007/s10472-019-09664-4.

5 Alexander Artikis, Anastasios Skarlatidis, François Portet, and Georgios Paliouras. Logic-based event recognition. *The Knowledge Engineering Review*, 27(4):469–506, 2012. doi:10.1017/S0269888912000264.

6 Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artif. Intell.*, 303:103649, 2022. doi:10.1016/j.artint.2021.103649.

7 João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733. IEEE Computer Society, 2017. doi:10.1109/CVPR.2017.502.

8 Ivan Donadello, Luciano Serafini, and Artur S. d'Avila Garcez. Logic tensor networks for semantic image interpretation. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1596–1602. ijcai.org, 2017. doi:10.24963/ijcai.2017/221.

9 Paolo Dragone, Stefano Teso, and Andrea Passerini. Neuro-symbolic constraint programming for structured prediction. In Artur S. d'Avila Garcez and Ernesto Jiménez-Ruiz, editors, *Proceedings of the 15th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 1st International Joint Conference on Learning & Reasoning (IJCLR 2021), Virtual conference, October 25-27, 2021*, volume 2986 of *CEUR Workshop Proceedings*, pages 6–14. CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2986/paper2.pdf.

10 José Roldán Gómez, Juan Boubeta-Puig, José Luis Martínez, and Guadalupe Ortiz. Integrating complex event processing and machine learning: An intelligent architecture for detecting iot security attacks. *Expert Syst. Appl.*, 149:113251, 2020. doi:10.1016/j.eswa.2020.113251.

11 Pascal Hitzler and Md Kamruzzaman Sarker. *Neuro-Symbolic Artificial Intelligence: The State of the Art*. IOS Press, 2022.

12 Jeya Vikranth Jeyakumar, Luke Dickens, Luis Garcia, Yu-Hsi Cheng, Diego Ramirez-Echavarria, Joseph Noor, Alessandra Russo, Lance M. Kaplan, Erik Blasch, and Mani B. Srivastava. Automatic concept extraction for concept bottleneck-based video classification. *CoRR*, abs/2206.10129, 2022. doi:10.48550/arXiv.2206.10129.

13 Abdullah Khan, Loris Bozzato, Luciano Serafini, and Beatrice Lazzerini. Visual Reasoning on Complex Events in Soccer Videos Using Answer Set Programming. In Diego Calvanese and Luca Iocchi, editors, *GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence, Bozen/Bolzano, Italy, 17-19 September 2019*, volume 65, pages 42–53. EasyChair, 2019. doi:10.29007/pjd4.

14 Abdullah Khan, Luciano Serafini, Loris Bozzato, and Beatrice Lazzerini. Event Detection from Video Using Answer Set Programing. In Alberto Casagrande and Eugenio G. Omodeo, editors, *Proceedings of the 34th Italian Conference on Computational Logic, Trieste, Italy, June 19-21, 2019*, volume 2396 of *CEUR Workshop Proceedings*, pages 48–58. CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2396/paper25.pdf.

15 Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 2020. URL: http://proceedings.mlr.press/v119/koh20a.html.

16 Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc D. Raedt. DeepProbLog: Neural Probabilistic Logic Programming. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, volume 31, pages 3753–3763, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/dc5d637ed5e62c36ecb73b654b05ba2a-Abstract.html.

17 Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. LYRICS: A General Interface Layer to Integrate Logic Inference and deep Learning. In Ulf Brefeld, Élisa Fromont, Andreas Hotho, Arno J. Knobbe, Marloes H. Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases – European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, volume 11907 of *Lecture Notes in Computer Science*, pages 283–298, 2019. doi:10.1007/978-3-030-46147-8_17.

18 Luc D. Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2462–2467, 2007. URL: http://ijcai.org/Proceedings/07/Papers/396.pdf.

19 Roberto Sebastiani and Silvia Tomasi. Optimization Modulo Theories with Linear Rational Costs. *ACM Transactions on Computational Logics*, 16(2), 2015.

20 Anastasios Skarlatidis, Alexander Artikis, Jason Filippou, and Georgios Paliouras. A probabilistic logic programming event calculus. *Theory and Practice of Logic Programming*, 15(2):213–245, 2015. doi:10.1017/S1471068413000690.

21 Praveen Tirupattur, Kevin Duarte, Yogesh Singh Rawat, and Mubarak Shah. Modeling multi-label action dependencies for temporal action localization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1460–1470. Computer Vision Foundation / IEEE, 2021. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Tirupattur_Modeling_Multi-Label_Action_Dependencies_for_Temporal_Action_Localization_CVPR_2021_paper.html, doi:10.1109/CVPR46437.2021.00151.

22 Marc Roig Vilamala, Liam Hiley, Yulia Hicks, Alun D. Preece, and Federico Cerutti. A pilot study on detecting violence in videos fusing proxy models. In *22th International Conference on Information Fusion, FUSION 2019, Ottawa, ON, Canada, July 2-5, 2019*, pages 1–8. IEEE, 2019. URL: https://ieeexplore.ieee.org/document/9011329.

23 Marc Roig Vilamala, Tianwei Xing, Harrison Taylor, Luis Garcia, Mani B. Srivastava, Lance M. Kaplan, Alun D. Preece, Angelika Kimmig, and Federico Cerutti. Using deepproblog to perform complex event processing on an audio stream. *CoRR*, abs/2110.08090, 2021. arXiv:2110.08090.

24 Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. Deepstochlog: Neural stochastic logic programming. *CoRR*, abs/2106.12574, 2021. arXiv:2106.12574.

25 Wei Xiang and Band Wan. A Survey of Event Extraction From Text. *IEEE Access*, 7:173111–173137, 2019. doi:10.1109/ACCESS.2019.2956831.

26 Tianwei Xing, Luis Garcia, Marc Roig Vilamala, Federico Cerutti, Lance M. Kaplan, Alun D. Preece, and Mani B. Srivastava. Neuroplex: learning to detect complex events in sensor networks through knowledge injection. In Jin Nakazawa and Polly Huang, editors, *SenSys '20: The 18th ACM Conference on Embedded Networked Sensor Systems, Virtual Event, Japan, November 16-19, 2020*, pages 489–502. ACM, 2020. doi:10.1145/3384419.3431158.

27 Tianwei Xing, Marc R. Vilamala, Luis Garcia, Federico Cerutti, Lance Kaplan, Alun Preece, and Mani Srivastava. DeepCEP: Deep Complex Event Processing Using Distributed Multimodal Information. In *IEEE International Conference on Smart Computing, SMARTCOMP 2019, Washington, DC, USA, June 12-15, 2019*, pages 87–92. IEEE, 2019. doi:10.1109/SMARTCOMP.2019.00034.

28 Zhun Yang, Adam Ishay, and Joohyung Lee. NeurASP: Embracing Neural Networks into Answer Set Programming. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1755–1762. ijcai.org, 2020. doi:10.24963/ijcai.2020/243.

29 Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *Int. J. Comput. Vis.*, 126(2-4):375–389, 2018. doi:10.1007/s11263-017-1013-y.

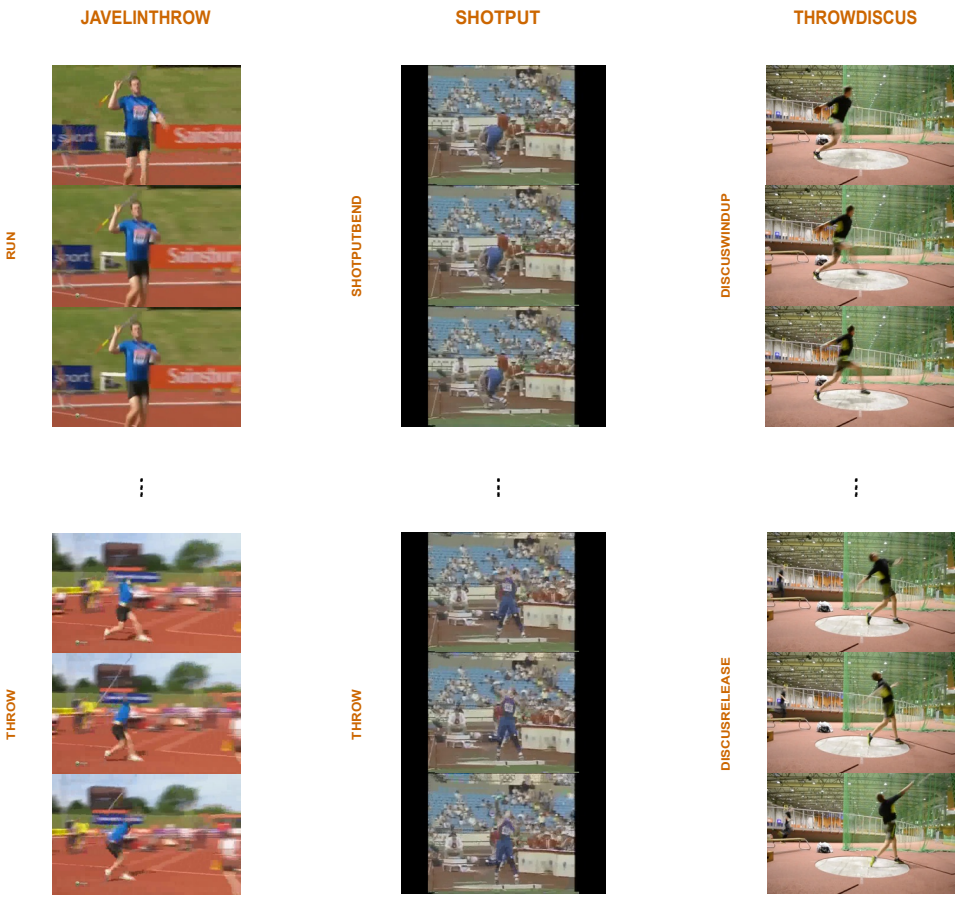

# A MILP for high jump

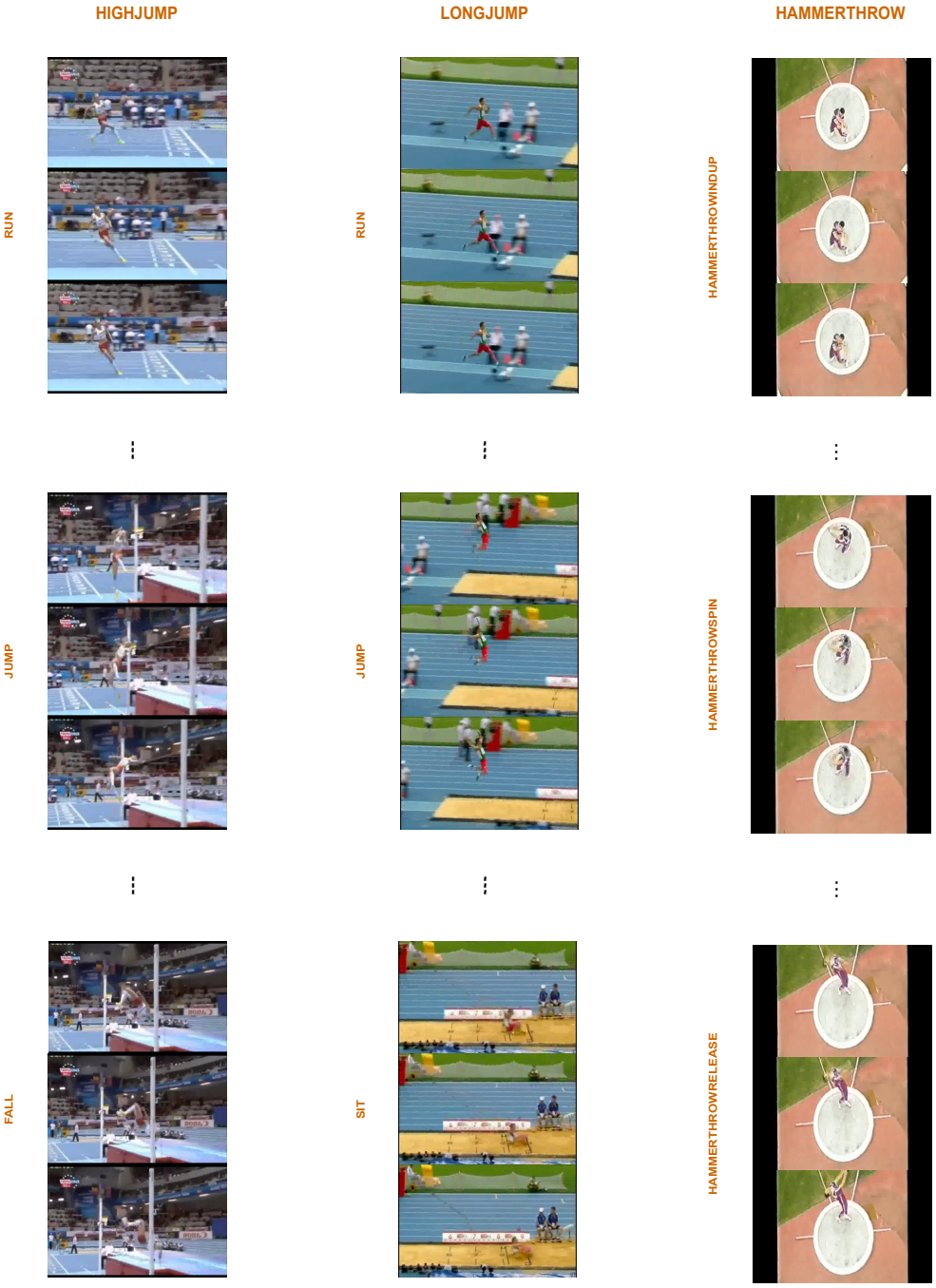
**Listing 1** Encoding high jump as a MILP using MiniZinc

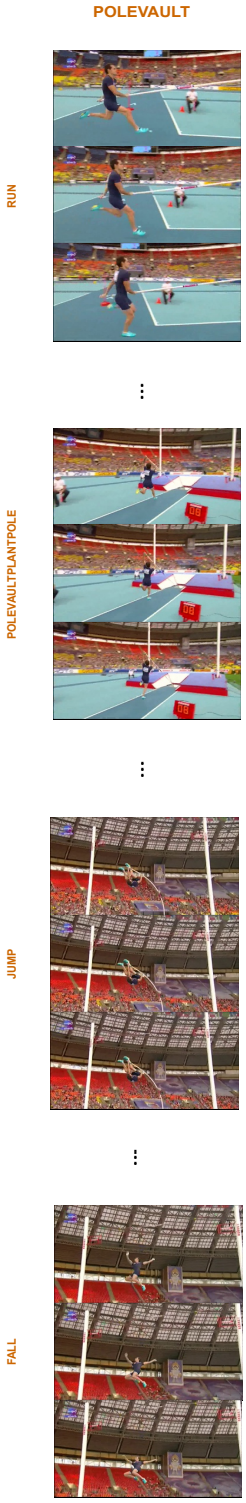
```
% HJ = High Jump
% R  = Run
% J  = Jump
% F  = Fall

int: bHJ;
int: eHJ;

int: minSum_R_J = 3;
int: maxSum_R_J = 48;
int: minSum_R_F = 4;
int: maxSum_R_F = 57;
int: minSum_J_F = 3;
int: maxSum_J_F = 33;

int: target_R_J = maxSum_R_J - minSum_R_J + 1;
int: target_R_F = maxSum_R_F - minSum_R_F + 1;
int: target_J_F = maxSum_J_F - minSum_J_F + 1;

var bHJ .. eHJ: bR;
var bHJ .. eHJ: eR;
var bHJ .. eHJ: bJ;
var bHJ .. eHJ: eJ;
var bHJ .. eHJ: bF;
var bHJ .. eHJ: eF;

var int: lenR = eR - bR + 1;
var int: lenJ = eJ - bJ + 1;
var int: lenF = eF - bF + 1;


constraint eR > bR /\ eJ > bJ /\ eF > bF;
constraint bR == bHJ /\ eR == (bJ-1) /\ eJ == (bF-1) /\ eF == eHJ;
constraint lenR >= (lenJ + lenF) /\ lenJ < (lenR + lenF) /\ lenF < (lenR + lenJ);


var int: cost_comp_run_pos = - sum (t in bR..eR) (ae_predictions[1,t]);
var int: cost_comp_run_neg = sum (t in (eR+1)..eHJ) (ae_predictions[1,t]);
var int: cost_comp_jump_pos = - sum (t in bJ..eJ) (ae_predictions[2,t]);
var int: cost_comp_jump_neg_1 = sum (t in bHJ..(bJ-1)) (ae_predictions[2,t]);
var int: cost_comp_jump_neg_2 = sum (t in (eJ+1)..eHJ) (ae_predictions[2,t]);
var int: cost_comp_fall_pos = - sum (t in bF..eF) (ae_predictions[3,t]);
var int: cost_comp_fall_neg = sum (t in bHJ..(bF-1)) (ae_predictions[3,t]);

var int: cost = (
  cost_comp_run_pos + cost_comp_run_neg
  + cost_comp_jump_pos + cost_comp_jump_neg_1 + cost_comp_jump_neg_2
  + cost_comp_fall_pos + cost_comp_fall_neg
  + 1000 * abs(target_R_J - (lenR + lenJ))
  + 1000 * abs(target_R_F - (lenR + lenF))
  + 1000 * abs(target_J_F - (lenJ + lenF))
);

solve minimize cost;
```

In lines 6-7, the constants representing the begin and the end of the clip are declared. These are going to be filled at running time and will change depending on the length of the clip processed. The blocks of lines 9-14 and 16-18 contains, the minimum/maximum sum of the length of the intervals of two atomic events and the target length in which the sum of the two intervals has to lie in. The declaration of the optimizer decision variables is contained in line 20-25. These variables are going to be set by Gurobi at the end of the optimization. In lines 27-29, the variables representing the length of the interval of each atomic events are defined. Lines 32-34 represent hard constraints that Gurobi has to satisfy. In details, line 32 states that the end of each atomic event has to be greater than their corresponding begin, while lines 33 and 34 defines respectively temporal relations among events and algebraic constraints among the length of the intervals of atomic events. In addition to the satisfaction of those constraints, Gurobi has to minimize a cost function (lines 45-62). The cost function can be split in two parts. The former (lines 46-48) is composed by the sum of components defined in lines 37-43 where we want to maximize (i.e. minimize) the sum of probability where the solver states the atomic events are happening (pos), and minimize (i.e. maximize) the sum of probability where the atomic events are not happening (neg). The latter (lines 49-51) refers to soft constraints where the solver has to set the the sum of the length of two atomic events' intervals to be as closed as possible to their target length.

## B Subset of structured and atomic events considered




HIGHJUMP
LONGJUMP
HAMMERTHROW
RUN
RUN
HAMMERTHROWINDUP
JUMP
JUMP
HAMMERTHROWSPIN
FALL
SIT
HAMMERTHROWRELEASE


POLEVAULT
RUN
POLEVAULTPLANTPOLE
JUMP
FALL

## C Temporal action localization model

The input of the model consists in a series of feature vectors $\boldsymbol{f}$ extracted by a pre-trained two-stream I3D [7], where each $f_i \in \boldsymbol{f}$ corresponds to 8 frames (or 0.33 seconds) and contains global information at both frame and video-clip level. A non linear transformation is applied on these features in order to obtain class level features ($C \times T \times H$) with $C$ representing the number of classes, $T$ the number of timestamps and $H$ the dimension of embedding space. Then, the class-level features are refined using $L$ attention-based Multi-Label Action Dependency (MLAD) layers. These layers are composed by two disjoint branches which adopt a self-attention operation to model the relationships between actions that happened within the same timestamp (referred as Co-occurence Dependency branch) and actions happening at different timestamps (referred as Temporal Dependency branch). As a result, a refined set of features is returned by each branch, respectively. At the end, a linear combination of the two branches' features is applied (through a learnt value $\alpha \in [0, 1]$) and the result is passed to $C$ individual classification layers which outputs class probabilities for each timestamp ($T \times C$).