Learning Asynchronous and Sparse Human-Object Interaction in Videos

Romero Morais*, Vuong Le, Svetha Venkatesh, Truyen Tran
Applied Artificial Intelligence Institute, Deakin University, Australia
{ralmeidabaratad,vuong.le,svetha.venkatesh,truyen.tran}@deakin.edu.au

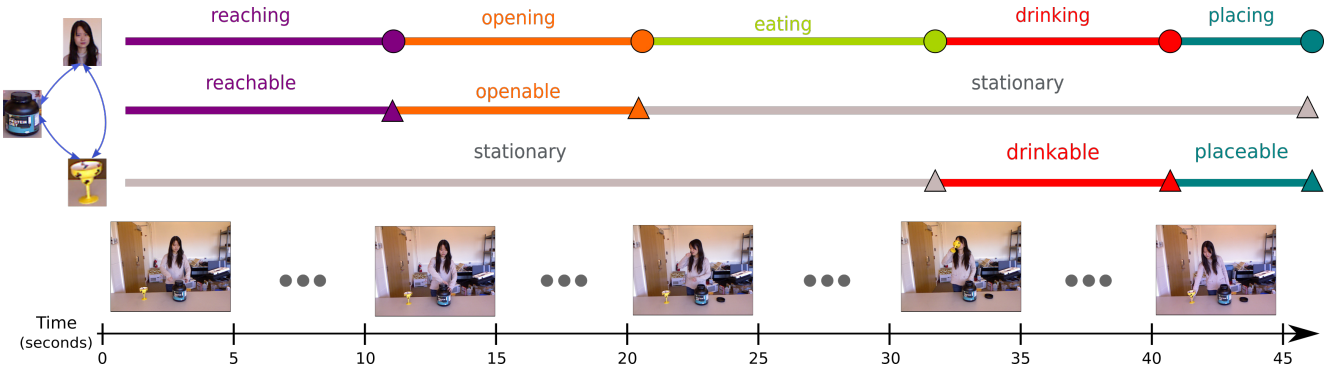

Figure 1: An example of human-object interaction activity, where a person takes some medicine and interacts with two objects. Human (circles) and object (triangles) entities have independent lives throughout the video (upper three rows). Although videos are captured in regular timing (lower rows), the dynamics of human activities and object affordances evolve sparsely and asynchronously with respect to each other (colored segments). They also affect each other (blue curved arrows). These characteristics of human-object interactions are the main modeling goals of this work.

Abstract

Human activities can be learned from video. With effective modeling it is possible to discover not only the action labels but also the temporal structure of the activities, such as the progression of the sub-activities. Automatically recognizing such structure from raw video signal is a new capability that promises authentic modeling and successful recognition of human-object interactions. Toward this goal, we introduce Asynchronous-Sparse Interaction Graph Networks (ASSIGN), a recurrent graph network that is able to automatically detect the structure of interaction events associated with entities in a video scene. ASSIGN pioneers learning of autonomous behavior of video entities including their dynamic structure and their interaction with the coexisting neighbors. Entities' lives in our model are asynchronous to those of others therefore more flexible in adapting to complex scenarios. Their interactions are sparse in time hence more faithful to the true underlying nature and more robust in inference and learning. ASSIGN is tested on human-object interaction recognition and shows superior performance in segmenting and labeling of human sub-activities and object affordances from raw videos. The native ability of ASSIGN in discovering temporal structure also eliminates the dependence on external segmentation that was previously mandatory for this task.

1. Introduction

Human activities are strongly connected to the surrounding environment and the objects in it. The interactions between human and object entities observed in videos are a fundamental clue toward a deep understanding of human behavior and the surrounding world [8]. This capability is reflected in the *human-object interaction (HOI) recognition* task, in which human sub-activities (such as *drinking*) and object affordances (such as *drinkable*) are segmented and recognized from a video by analyzing the interactive relations between entities (Fig. 1). These relations naturally form a spatio-temporal graph where entities (humans or objects) and their dynamic interactions evolve throughout the activity. Although entities can be detected and tracked from video, it is challenging to build a graph model that can automatically discover the temporal structure of activities and natively reflect the complex and intricate nature of these interactions.

Currently available approaches applied conditional random fields [14, 17, 18] and graph neural networks [6, 30] to model the spatio-temporal entity interaction graph. These models assume knowledge of the temporal structure of the video and are limited to the task of assigning activity and affordance labels to the segments.

Rather than this cascaded approach, we exploit the fact that structure and content of events are tightly coupled and may support each other toward the optimal solution in a

joint discovery scheme. Such scheme further allows to break the common assumption that entities in a video are always active and interact continuously. In reality, unlike regularly captured video frames, interactions between entities happen sparsely in time. This suggests that temporal relations in the interaction graph can be pruned into a more concise and efficient graph structure. Authentic modeling of the asynchronous lives of entities allows them to act independently and only update their state when needed.

In light of that, we introduce Asynchronous-Sparse Interaction Graph Networks (ASSIGN), a joint structure-content discovery framework for sparse and asynchronous human-object interactions. ASSIGN stands on the principle that each entity has an independent life in a video, where each entity behaves and interacts with its coexisting neighbors at its own pace and timing. The temporal structure and the labels of events are discovered jointly using a flexible two-layer dynamic graph network, that can do inference and be trained end-to-end without dependence on external temporal segmentation of events.

We demonstrate the segmentation and labeling capabilities of ASSIGN on two major human-object interaction datasets, where ASSIGN attains superior quantitative performance and more realistic qualitative results when compared to related methods.

In summary, this paper makes three major contributions:

- Constructs the first end-to-end graph model that jointly learns temporal structure and content label of human-object interaction activities;
- Effectively models the sparse and asynchronous entities lives in the context of a social activity; and
- Permits efficient relational inference that can skip unnecessary operations, which results in increased robustness to a wide variety of event structures.

2. Related Work

2.1. Human-object interaction in videos

Traditional approaches to HOI modeling in videos surround on variations of Markov Random Fields (MRF). Koppula *et al*. [17] used an MRF to model entities in videos with fully connected spatial and temporal edges. It also starts a trend to use sub-activity segments as temporal time units. The work of Koppula *et al*. [17] is extended into the ATCRF model [18], which anticipates future sub-activity/affordances and gathers features from frame-level nodes. ATCRF is further advanced into GP-LCRF [14] to reduce the dimensionality of the frame-level human representation. Another extension of ATCRF is the Recursive CRF [33], in which the CRF is placed under a Bayesian filtering with an efficient belief computation. With the recent advancement of spatio-temporal relation modules, MRF-like models advanced into more efficient implementations with recurrent neural networks (RNNs) and graph neural networks (GNNs). Jain *et al*. [12] proposed to factorize the dynamic relationships in HOI and model the factors with a mixture of RNNs. Qi *et al*. [30] proposed Graph Parsing Neural Network (GPNN), which allows spatial graph topology to be inferred adaptively. Ghosh *et al*. [6] extended GNNs with a stacked hourglass network [27] for label prediction. The MRF and GNN families of models are reliable in predicting HOI labels but they cannot perform temporal segmentation themselves and need to resort to an external segmentation method (such as dynamic programming) either before or during inference. ASSIGN, on the other hand, learns the segmentation in tandem with the labeling directly from frame-level features.

This joint capability is also the goal of several efforts to constrain HOIs with activity grammars borrowed from natural language processing, namely Stochastic grammar [28] and Earley tree parser [29]. These constraints improve the ability to learn relationships between the entities via an explicit regulating grammar but at the same time limits their flexibility in the process. ASSIGN is less regulated than grammar-based approaches, but is less sensitive to noise and more scalable to the size of the problem. We view ASSIGN as a complementary approach that can also be further integrated with grammars.

Shared between all previous works is the assumption that entities are synchronous and constantly update their states. This oversimplification is unnatural and a source of practical issues such as over-segmentation. In this work, we directly challenge this assumption by modeling the independent behavior and sparse interactions between the entities.

2.2. Action segmentation

Action segmentation is another line of work that shares with us the goal of finding temporal structure of activities in video. Notable works include semi-Markov model [35], multi-class SVM on spatial bag-of-words [9], and segmental RNNs [16]. More recently, CNN methods dominate the action segmentation literature [20, 21, 22]. Farha and Gall [5], for instance, proposed a multi-stage CNN (MS-TCN) with and a truncated MSE loss to handle over-segmentation. TransParser [34] learns the temporal segmentation without supervision from sub-action labels with local and global losses. Unlike this line of work where activities are singularly modeled, we explore the interactions between human and objects and consider the relations between human sub-activities and object affordances throughout the video.

2.3. Sparse and asynchronous event modeling

The sparsity and asynchronicity of events have been a modeling goal of the signal processing community. Neil

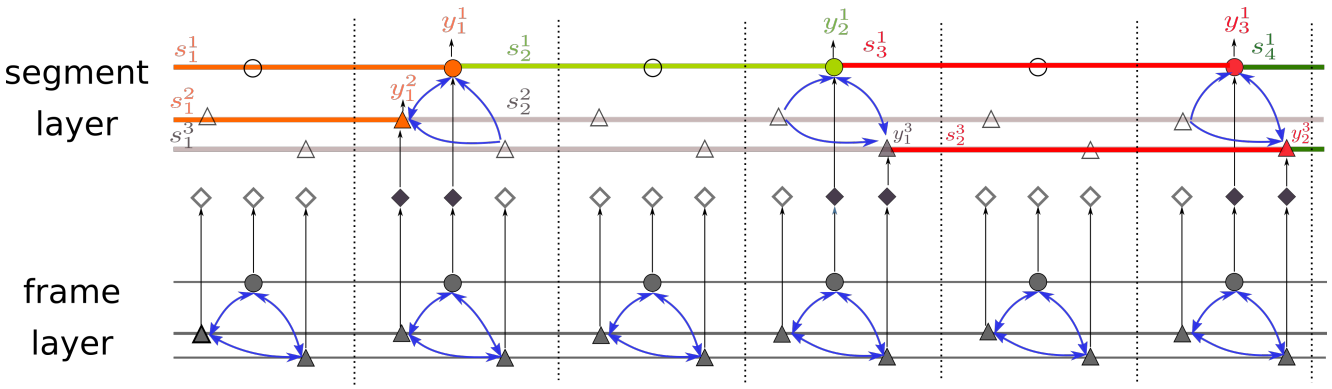

Figure 2: Asynchronous-Sparse Interaction Graph Networks (ASSIGN) architecture contains two layers of spatio-temporal graph networks. At each layer, graph nodes represent human (circle) or object (triangle) entities. Spatial edges are modeled with message passing (blue curved arrows), and temporal edges are modeled with recurrent networks (horizontal lines). The frame level of ASSIGN updates at every time step, for every entity, and decides at each step (upward arrows) whether the corresponding segment-level entity *changes* state (solid diamond) or *skips* an update (hollow diamond)—details in Sec. 3.3. The sparse *change* signals lead to asynchronous and sparse updates (solid shapes) and interactions (blue curved arrows) at the segment level of ASSIGN—details in Sec. 3.4. Segment labels are generated by the second layer at the update operators.

et al. [26] extended the long short-term memory (LSTM) cell [10] formulation with a "time" gate, which introduces "open" and "close" cycles, to allow sparsity in the state updates. Similarly, Campos *et al*. [2] introduced sparsity into RNN updates by learning binary decisions regularized by a budget loss to skip redundant state updates. In contrast to these works, we skip state updates not only to reduce the computational complexity but also to match the semantics of the human activities. Furthermore, ASSIGN is more advanced in fully using the dense input signal for the sparse activity decisions.

In processing naturally sparse signals, Sekikawa *et al*. [32] proposed EventNet for real-time asynchronous event streams from event-based cameras. EventNet process events via a two-module architecture timed by the input events and output predictions. Asynchronous data from event-based cameras are also handled by an extended version [24] of the Submanifold Sparse Convolutions (SSC) [7], which extend the spatial sparsity modeling of SSC with localized updates throughout the convolutional maps by keeping track of a rulebook per layer. The key difference between this line of work and our formulation is that we explore sparse information from dense signals instead of assuming the signal is already sparse.

3. Method

3.1. Problem formulation

We are interested in learning the spatio-temporal structure of human-object interactions (HOI) in videos. Previous works consider special cases of a single human [18, 30] or two human hands interacting with multiple objects [4]. We approach this problem in a generic way, where we model an arbitrary number of humans and objects in a video. The problem is defined on a video of T frames with N entities (humans and objects) in it. These entities have their features extracted by detecting and tracking them throughout the video. The e-th entity is represented by a temporal sequence of frame-level features $X^e = \{x_t^e\}_{t=1...T}$ together with a class label c^e. In human-object interaction, this label holds the value of either *human* or *object*.

A HOI recognition problem is defined to use the input $\{X^e, c^e\}_{e=1...N}$ to generate a *temporal segmentation* for each entity. For the e-th entity, the segmentation is of the form $S^e = \{s_1^e, s_2^e, \ldots, s_{n_e}^e\}$, where the k-th member segment is represented by its start time and end time (which is the start time of next segment) $s_k^e = [t_k^e, t_{k+1}^e)$. The output also includes the prediction of segment labels $y^e = \{y_1^e, \ldots, y_{n_e}^e\}$ which effectively are sub-activity labels for humans and affordance labels for objects.

For a human entity, a segment label is the name of a sub-activity, and for an object entity it is the name of an affordance. These labels are interlinked with each other; for example, a human sub-activity *drinking* usually overlaps with affordance *drinkable* of a cup. But, they do not need to be perfectly aligned. An object can remain in the same state during all human activities not involving it. Modeling these sparse and asynchronous relations is the goal of this work.

3.2. Asynchronous-Sparse Interaction Graph Network

We aim at learning the temporal segmentation and label of sparse events associated with asynchronous entities in a video. To this end, we design a two-layer asynchronous recurrent graph network named Asynchronous-Sparse Interaction Graph Network (ASSIGN). ASSIGN is specialized

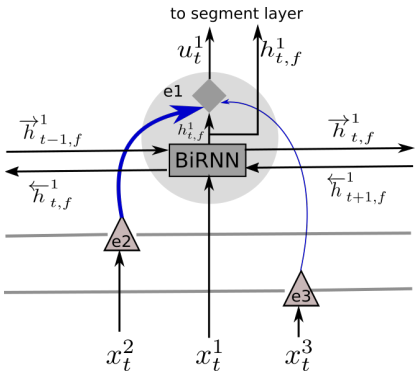

Figure 3: *Frame-level* node (only detailed for human node e1) with BiRNN unit (rectangle) and *segment boundary detector* (diamond shape). The detector considers the current recurrent state and messages from neighboring nodes (blue curved arrows) weighted by an attention mechanism (thickness of arrows). It then makes a decision u_t^e, for each entity, on whether frame t is the final frame of a segment or not. If it is a positive signal ($u_t^e = 1$), the summarized context $h_{t,f}^1$ is sent up to the segment-level node to predict the label of the finished segment and start a new one.

in modeling each entity in a video with two spatio-temporal graphs, one at frame level and one at segment level (Fig. 2). The frame-level graph nodes process video frames and update their states at every time step, whereas the segment-level graph nodes update sparsely—only when signaled by the frame-level partner to do so. Each entity decides its own pace asynchronously with consideration to its neighbors.

3.3. Segmenting the entity life

The primary task for ASSIGN is to learn the temporal segmentation of every entity in a video. This translates into making a binary decision at each time step of whether the current segment ends and a new segment starts or not. The segment change of a sub-activity, or affordance, depends on the internal state of the entity in question and its relation with its neighbors. For example, a human that gets close to a cup makes it a *drinkable* object. This insight is realized into the design of the frame-level layer of ASSIGN (Fig. 3).

The frame-level layer of ASSIGN takes as input $\{X^e, c^e\}_{e=1\ldots N}$ and builds a spatio-temporal graph. Spatial edges represent interactions between entities, and temporal edges connect instances of the same entity throughout time and represent the internal progression of such an entity. We implement temporal edges as Bidirectional RNNs (BiRNN) and generate the hidden state of the e-th entity at the t-th frame by

$$h_{t,f}^e = \text{BiRNN}_f\left(x_t^e, \overrightarrow{h}_{t-1,f}^e, \overleftarrow{h}_{t+1,f}^e\right), \qquad (1)$$

where $\overrightarrow{h}_{t-1,f}^e$ and $\overleftarrow{h}_{t+1,f}^e$ are forward and backward RNN states, and $h_{t,f}^e$ is the concatenated output of the two RNNs: $h_{t,f}^e = \left[\overrightarrow{h}_{t,f}^e, \overleftarrow{h}_{t,f}^e\right]$.

The spatial edges connect different entities at the same time step and reflect the dynamic relations of neighboring entities. It is implemented by pair-wise messaging between entities, and we distinguish between two types of spatial messages: (1) *intra-class messages* from entities of the same class and (2) *inter-class messages* from entities of different classes. This distinction is important because the nature of the relations are different. The collaboration between two human entities, for example, must be modeled differently than the effect of an object on a human.

The frame-level inter-class message to entity e at time t is calculated by:

$$m_{t,f}^{\text{inter}\rightarrow e} = \text{Att}\left(\left[x_t^e, h_{t,f}^e\right], \left\{\left[x_t^k, h_{t,f}^k\right]\right\}_{c^k \neq c^e}\right). \qquad (2)$$

Here, Att is the attention operator that calculates a weighted average of the contributions from the neighboring entities. In ASSIGN, it is implemented by a variant of scaled dot-product attention [36] with identical keys and values

$$\text{Att}\left(q, \{v_i\}_{i=1\ldots n}\right) = \sum_{i=1}^{n} \text{softmax}\left(\frac{q^T v_i}{\sqrt{d}}\right) v_i, \qquad (3)$$

where q is a query vector, $\{v_i\}$ is a set of keys/values vectors of size n and, and d is the feature dimension.

Effectively, this operation combines the hidden states $h_{t,f}^*$ and inputs x_t^* of the entities and use them as both keys/values and queries in weighing the relevance of the interacted neighboring nodes (blue arrows in Fig. 3).

Similarly, the intra-class message is calculated on the set of entities from the same class:

$$m_{t,f}^{\text{intra}\rightarrow e} = \text{Att}\left(\left[x_t^e, h_{t,f}^e\right], \left\{\left[x_t^k, h_{t,f}^k\right]\right\}_{k \neq e, c^k = c^e}\right). \qquad (4)$$

These spatial edges resemble a graph attention network [37], except that they dynamically evolve through time.

Finally, we gather the current temporal recurrent state together with the spatial relation messages to make the segmenting decision. This is done by the *segment boundary detector* (diamond shape in Fig. 3). It contains an MLP γ and a differentiable discrete valued estimator using the Gumbel-Softmax (GSM) operator [13, 23]:

$$u_t^e = \text{GSM}\left\{\gamma\left(\left[x_t^e, h_{t,f}^e, m_{t,f}^{\text{intra}}, m_{t,f}^{\text{inter}}\right]\right)\right\}. \qquad (5)$$

The binary output $u_t^e = 1$ indicates that t is the last frame of the current segment for the e-th entity and $u_t^e = 0$ otherwise. This segmenting signal controls the behavior of the segment-level nodes, which we describe next.

3.4. Labeling the learned segments

The segment layer of ASSIGN manages the spatio-temporal dynamics of the segments whose boundaries are provided by the frame layer via the segmenting signal u_t^e and frame-level state $h_{t,f}^e$. This layer is also modeled as a spatio-temporal graph with BiRNN for temporal edges and attentional message passing for spatial connections, similarly to the frame layer.

The key specialty of this graph layer is that its operations are not dense and regular as in the frame layer. Each entity can either update or copy its state, depending on the provided signal u_t^e. This adaptive operation constitutes the asynchronous and sparse behavior of ASSIGN.

At each time step t, if $u_t^e = 1$, the node gathers information from the context and update its state using its recurrent operator (blue arrows in the upper half of Fig. 2). This includes the segment-level inter-class message

$$m_{t,s}^{\text{inter}\to e} = \text{Att}\left(h_{t-1,s}^e, \left\{h_{t-1,s}^k\right\}_{c^k \neq c^e}\right), \quad (6)$$

and intra-class message

$$m_{t,s}^{\text{intra}\to e} = \text{Att}\left(h_{t-1,s}^e, \left\{h_{t-1,s}^k\right\}_{k\neq e, c^k = c^e}\right), \quad (7)$$

where Att is defined in Eq. 3, and these messages are calculated similarly to frame-level counterparts in Eqs. 2 and 4. The main distinction is that they are calculated sparsely, only when needed.

We combine these segment-level messages with the frame-level state $h_{t,f}^e$ and messages $m_{t,f}^{\text{inter}\to e}$ and $m_{t,f}^{\text{intra}\to e}$, previously calculated by the frame layer, to form the segment-level feature z_t^e:

$$z_t^e = \left[h_{t,f}^e, m_{t,f}^{\text{inter}\to e}, m_{t,f}^{\text{intra}\to e}, m_{t,s}^{\text{inter}\to e}, m_{t,s}^{\text{intra}\to e}\right]. \quad (8)$$

This input is fed to the segment-level BiRNN units (BiRNN_s) to update their states:

$$h_{t,s}^e = \text{BiRNN}_s\left(z_t^e, \overrightarrow{h}_{t-1,s}^e, \overleftarrow{h}_{t+1,s}^e\right). \quad (9)$$

The updated state is then used to recognize the label of the finished segment

$$\hat{y}_t^e = \text{Softmax}\left(\sigma(h_{t,s}^e)\right), \quad (10)$$

where σ is an MLP and the Softmax is calculated over the appropriate label set, either human sub-activities or object affordances.

In the other case where $u_t^e = 0$, the node skips a BiRNN update and maintain its current state. This contextualized skipping not only creates sparsity in the state updates but also in the interactions. The inward messages are skipped while the outward messages to other updating neighbors can still happen.

This is a better reflection of the world where at-rest entities (*e.g.* objects far from humans) can avoid unnecessary state updates and over-segment predictions. It also prevents the short-term memory of the RNNs from fading quickly. Furthermore, operating at the segment-level separates semantic progress (activity) from raw signals (frames), thus it is more robust to varied video sampling rates. The sophisticated architecture of ASSIGN requires a customized training procedure, which we describe in the next section.

3.5. Model training

ASSIGN is effectively a multi-task learning framework where segmentation and labeling tasks are trained together in an end-to-end fashion. It is therefore trained by an ensemble of two losses for the two tasks.

For segmentation, we minimize the binary cross-entropy between a smoothed version of the ground-truth segmentation and the soft output of the boundary detector in Eq. 5

$$\mathcal{L}_{\text{Seg}} = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{N}\sum_{e=1}^{N}\text{BCE}\left(\hat{\tilde{u}}_t^e, \tilde{u}_t^e\right)\right], \quad (11)$$

where $\hat{\tilde{u}}_t^e$ is the real value of u_t^e before binary thresholding, and $\tilde{u}_t^e$ is the smoothed version (with Gaussian filter of $\sigma = 4$) of the binary pulse ground-truth segmentation.

For labeling, we minimize the negative log-likelihood of the predicted sub-activities and affordance labels:

$$\mathcal{L}_{\text{Label}} = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{N}\sum_{n=1}^{N}\text{NLL}\left(\hat{y}_t^n, y_t^n\right)\right]. \quad (12)$$

Even though the label is predicted per segment, this loss is calculated per frame so that long segments contribute more than short ones. The overall loss is the weighted sum of the two losses

$$\mathcal{L} = \mathcal{L}_{\text{Label}} + \lambda\mathcal{L}_{\text{Seg}}, \quad (13)$$

where λ is a tunable parameter.

While the sparsity of ASSIGN's operations provide a strong advantage in authentic modeling, it creates a subtle obstacle in training where informative gradients from the labeling loss in the segment layer rarely reaches the frame layer. To overcome this issue, we use a two-stage training procedure. In stage 1, we switch off $\mathcal{L}_{\text{Seg}}$ and set $u_t^e := 1$ everywhere so that the frame layer receives a constant stream of directive signals. In stage 2, we turn on the full model and continue to train on top of parameters learned in stage 1. We observed in our experiments that this two-stage training leads to faster convergence and improved final results.

3.6. Implementation details

For entity features, we use 2048-dimensional ROI pooling features extracted from the 2D bounding boxes of humans and objects in the video detected by a Faster R-CNN

[31] module pre-trained [1] on the Visual Genome dataset [19]. We optimize model parameters using the ADAM optimizer [15], with a learning rate of 10^{-3}. All recurrent networks are built with Gated Recurrent Units (GRU) [3]. The videos are resampled to a uniform 10 FPS frame rate before feeding to the frame-level network. For model selection, we use 10% of the training data as validation data and select the model with the lowest validation loss.

4. Experiments

4.1. Datasets

We evaluate ASSIGN on the CAD-120 [17] and on the Bimanual Actions [4] datasets. CAD-120 is the most popular dataset for HOI recognition. It contains 120 RGB-D videos of 4 subjects executing 10 different activities, each activity repeated 3 times. Each video depicts a single person interacting with 1–5 objects. In total, there are 10 human sub-activities and 12 object affordances, and each entity is annotated frame-wise.

We also experiment on the Bimanual Actions, the first HOI dataset of activities featuring a subject using both hands to interact with objects (*e.g*. left hand holds a nail while right hand hits the nail). It has 540 RGB-D videos of 6 subjects conducting 9 different tasks, each task repeated 10 times. The actions of each hand are annotated frame-wise as one of 14 possible actions. For both datasets, we only use the RGB channels to extract frame features.

4.2. Experimental settings

We evaluate ASSIGN on two tasks: joint segmentation and label recognition, and label recognition with known segmentation. The first task requires models to segment the time line for each entity in a video and label those segments. The second task is a special case of the first one where the ground-truth segmentation is known and models only need to label the provided segments.

To evaluate how well ASSIGN generalizes to unseen subjects, we do leave-one-subject out cross-validation on both datasets. Previous works focused on recognition of labels and usually report frame-level F_1 scores. These metrics, however, are not optimal for tasks involving segmentation because a method might heavily over- or under-segment a video and still attain reasonable frame-level scores. To amend that, we use the $\mathrm{F}_1@k$ metric [20] for the commonly used values of $k = 0.10, 0.25,$ and 0.50. The $\mathrm{F}_1@k$ metric considers a predicted segment correct if its IoU with the ground-truth segment is at least k. Wrong predictions and missed ground-truth segments count as false positives and false negatives, respectively. The $\mathrm{F}_1@k$ is a superior choice over frame-based metrics for joint segmentation and labeling problems and is widely used in previous segmentation works [5, 20, 25]. Note that for the label recognition with known segmentation task, $\mathrm{F}_1@k$ is constant for any k and reduces to segment-level micro and macro F_1 scores. We report these metrics, in line with other reported results in the literature.

4.3. Quantitative results

4.3.1 Joint segmentation and label recognition

In this main experiment, we compare the performance of ASSIGN with related state-of-the-art methods and two BiRNN-based baselines on the joint segmentation and label recognition task on the CAD-120 dataset. For this task, the input must be raw video features, with no trace of preproduced segmentation.

Two previous works fully qualify for this task: ATCRF [18] and rCRF [33]. Other major related works used the preproduced segmentation information in either explicit or implicit ways. Stochastic grammar [28] used the statistics of segmentation from test portion in training. Earley tree parser [29] repeats the preproduced segment-level features as frame-level features hence implicitly acknowledging the true segment boundaries. More concrete details about these uses are included in the Supplementary material.

The baselines are two variations of BiRNN GRU: The *Independent BiRNN* models each entity independently (*i.e*. no spatial messages), and the *Relational BiRNN* adds dense spatial interactions between entities. Further details are in the Supplementary material.

We show the $\mathrm{F}_1@k$ results in Table 1 and the frame-level F_1 in the Supplementary material. ASSIGN outperforms both the state-of-the-art methods and baselines in every configuration of the $\mathrm{F}_1@k$ measure for both human sub-activities and object affordances.

These results showcase the advantages of jointly segmenting and labeling. Other methods employ separate segmentation and labeling steps, and generate their final result by voting on many different segmentation options. This strategy has a weakness of increasing the over-segmentation when voters disagree and inability to correct if they make the same mistakes.

The BiRNN baselines make frame-wise predictions and lack relational modeling, thus they do not fully leverage the human-object interactions. Despite being simpler, the Independent BiRNN is superior to the Relational BiRNN with respect to object affordances. This can be explained by the infrequent changes of object affordances that were mistaken by the presence of dense messages from the human nodes in the Relational BiRNN. In contrast, ASSIGN allows sparse messaging and effectively overcomes these problems.

4.3.2 Label recognition only

To examine the sole capability of predicting labels, and to match with the task done by more previous works, we set

Table 1: *Joint segmentation and label recognition* task with no pre-segmentation. Performance on the CAD-120 dataset.

Model	Sub-activity			Object Affordance		
	F_1@0.10	F_1@0.25	F_1@0.50	F_1@0.10	F_1@0.25	F_1@0.50
rCRF [33]	65.6 ± 3.2	61.5 ± 4.1	47.1 ± 4.3	72.1 ± 2.5	69.1 ± 3.3	57.0 ± 3.5
Ind. BiRNN	70.2 ± 5.5	64.1 ± 5.3	48.9 ± 6.8	84.6 ± 2.1	81.5 ± 2.7	71.4 ± 4.9
ATCRF [18]	72.0 ± 2.8	68.9 ± 3.6	53.5 ± 4.3	79.9 ± 3.1	77.0 ± 4.1	63.3 ± 4.9
Rel. BiRNN	79.2 ± 2.5	75.2 ± 3.5	62.5 ± 5.5	82.3 ± 2.3	78.5 ± 2.7	68.9 ± 4.9
ASSIGN	**88.0** ± 1.8	**84.8** ± 3.0	**73.8** ± 5.8	**92.0** ± 1.1	**90.2** ± 1.8	**82.4** ± 3.5

Table 2: *Label recognition only* task with ground-truth segmentation. Performance on the CAD-120 dataset. Unreported results are marked as "-". An "*" means we reproduced results to match our *leave-one-subject out* protocol.

Model	Sub-activity F_1 (%)		Object Aff. F_1 (%)	
	Micro	Macro	Micro	Macro
GPNN* [30]	76.6	72.7	74.6	54.1
S-RNN [12]	82.4	-	91.1	-
KGS [17]	86.0	80.4	91.8	81.5
Lat. Linear-CRF [11]	87.0	86.0	-	-
ATCRF [18]	89.3	86.4	93.9	85.7
STGCN [6]	-	87.2	-	-
ASSIGN	**89.9**	**87.8**	**95.9**	**91.9**

Table 3: *Joint segmentation and label recognition* task with multiple human entities. Performance on Bimanual Actions dataset.

Model	Sub-activity		
	F_1@0.10	F_1@0.25	F_1@0.50
Dreher *et al*. [4]	40.6 ± 7.2	34.8 ± 7.1	22.2 ± 5.7
Ind. BiRNN	74.8 ± 7.0	72.0 ± 7.0	61.8 ± 7.3
Rel. BiRNN	77.7 ± 3.9	75.0 ± 4.2	64.8 ± 5.3
ASSIGN	**84.0** ± 2.0	**81.2** ± 2.0	**68.5** ± 3.3

up a simpler experiment where the true segmentation is provided to all methods. This skips the segmentation functionality of ASSIGN and put it in fair comparison with all previous works in their common experimental protocol.

Table 2 shows the micro and macro F_1 accuracy on CAD-120 dataset. For both metrics and for both sub-activity and object affordance, ASSIGN outperforms all other methods. This further demonstrates that our modeling of entities with asynchronous and sparse interactions is a more correct way to label the segments, agnostic to the segmentation quality.

4.3.3 Multiple human entities

The generic formulation of ASSIGN allows it to be easily applied to a wide range of scenarios. Thus, we trial ASSIGN in the case where multiple humans jointly do a task. This experiment is done on the Bimanual Actions dataset, which contains activities where a person's hands interact with many objects (we treat the hands as the multiple humans). We compare ASSIGN to the BiRNN baselines (Sec. 4.3.1) and the method of Dreher *et al*. [4], which is the only previous work proposed for this multi-human setting.

Table 3 compares the performance of these methods on the joint segmentation and labeling task. Dreher *et al*. [4] has the weakest performance, and this can be attributed to their over-simplistic graph network, which ignores the interactions between the hands and does not take long-term temporal context into account. The BiRNN baselines improve over Dreher *et al*. [4] by considering longer temporal context but fall short in reaching high accuracy for not modeling human-human interactions. ASSIGN makes major improvements over these methods by incorporating cross-hand spatial interaction and asynchronous long-term temporal context. Our higher performance is also attributed to the segment-level label decisions in contrast to frame-based decisions of the baselines.

Through the quantitative experiments, it is clear that joint structure-content exploration with consideration to the temporal life of entities are key features of ASSIGN that makes it excel in recognition performance. Next, we examine qualitative results and internal operations of ASSIGN.

4.4. Qualitative analysis

We compare the outputs of ASSIGN and related methods on examples from CAD-120 and Bimanual Actions datasets. Figure 4 shows an example in CAD-120 where ATCRF over-segments human sub-activities. Also, because segments are synchronized between entities, these errors spread to objects and hurt the accuracy on affordance recognition. ASSIGN, on the other hand, successfully overcomes the over-segmentation and the error propagation by supporting sparse and asynchronous processing.

Figure 5 shows an example of a *cooking* task on the Bimanual dataset. Dreher *et al*. [4] take limited temporal context into account and creates many short segments. Relational BiRNN improves on short segments but fails to handle long ones (*e.g*. long *stir* action of the right hand). ASSIGN, with more advanced modeling, is well equipped to reliably handle both short- and long-term interactions.

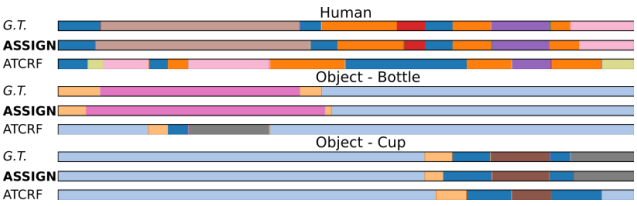

Figure 4: Segmentation and labeling results for the ASSIGN and ATCRF methods compared to ground-truth on the CAD-120 dataset for a *taking medicine* activity. In this example, ATCRF over-segments the long *opening* ( ) sub-activity for the human. Because objects are synchronized with human in ATCRF, these over-segmentations creates a domino effect that leads to the incoherent structure of the Bottle timeline. In contrast, ASSIGN allows asynchronous state changes of human and objects and avoids this type of mistake. Legend: Sub-activities - *reaching,* *opening,* *moving,* *eating*, *drinking*, *placing*, and *null*; Affordances - *reachable*, *openable,* *stationary*, *movable*, *drinkable*, and *placeable*.

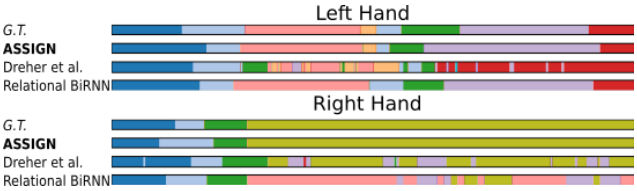

Figure 5: Segmentation and labeling results on the Bimanual dataset for a *cooking* task. In this example, Dreher *et al*. [4] create many spurious short segments due to their model's limited temporal context. The Relational BiRNN baseline improves on short sub-activities but fails to handle longer events such as the long *stir* ( ) action because the recurrent memory forgets quickly. On the other hand, ASSIGN handles long actions well by appropriately skipping redundant updates. Legend: *idle*, *approach*, *lift*, *stir*, *hold,* *retreat*, *pour*, and *place*.

In Figure 6, we analyze the attention scores of the objects in relation to the human at both levels of ASSIGN. At the frame level the human pays sharper attention to a specific object in order to make a clean decision on transitioning between sub-activities. At the segment level the attention is more uniform, which is reasonable given the sparsity of the updates. At each sparse deciding point, the human needs to consider multiple neighboring objects to recognize the label of its sub-activity.

4.5. Ablation study

To understand the role of individual components of ASSIGN, we ablate several key modules and evaluate these variants on the CAD-120 dataset (Table 4). First, the spatial message passing has a crucial role in modeling the entity

Table 4: Ablation study on the CAD-120 dataset.

	Model	Sub-activity		Object Affordance	
		F_1@0.10	F_1@0.50	F_1@0.10	F_1@0.50
1	w/o msg passing	74.5	55.0	89.0	74.4
2	w/o seg. loss	84.3	69.9	89.2	78.6
3	w/ dense update	85.5	70.3	90.6	79.8
4	w/o pre-training	87.6	71.6	91.1	78.9
5	full ASSIGN model	**88.0**	**73.8**	**92.0**	**82.4**

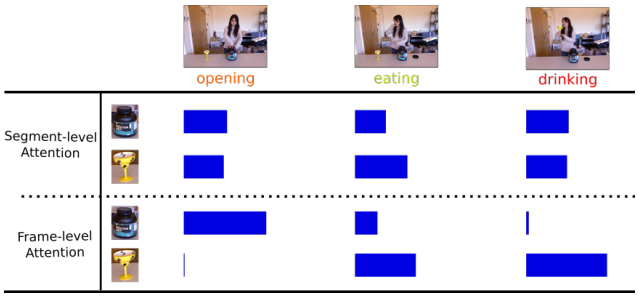

Figure 6: Attention scores of messages from objects to human at both layers. Sharp and strong attention on relevant objects are used in frame level to gather key information. More uniform attention is found in the segment level where overall consideration is made.

interactions (row 1). Second, we join labeling and segmenting tasks by adding the segmentation loss, which is also essential for good joint results (row 2). On top of this joint training scheme, ASSIGN is special by using asynchronous and sparse interaction constraints. This innovation significantly improves robustness to a wide variety of activity structures (row 3 vs. row 5). Finally, the strategy of pre-training ASSIGN with the dense model (see Sec. 3.5) benefits the learning process and supports the model to reach the highest performance (row 4 vs. row 5).

5. Conclusions

We designed ASSIGN, a two-layer graph network that explores the activity structure concurrently with predicting its content. ASSIGN models human-object interaction more correctly than previous methods by allowing the participating entities to have asynchronous lives. The interactions in ASSIGN are sparse, hence more robust to varied segment lengths and activity progression.

These advantages resulted in higher performance in multiple datasets. Moreover, this performance is consistently strong over larger variations of scenarios than any other method. Deep analysis into ASSIGN's operation shows that the strong performance comes from the new ability to deal with over- or under-segmentation mistakes that previous models suffered from. The generic capability of distilling structure from time series showcases that ASSIGN can be readily applicable to other domains and applications.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-Up and Top-Down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, June 2018. 3.6

[2] Víctor Campos, Brendan Jou, Xavier Giró-i Nieto, Jordi Torres, and Shih-Fu Chang. Skip RNN: Learning to skip state updates in recurrent neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, February 2018. 2.3

[3] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. ACL, 2014. 3.6

[4] Christian R G Dreher, Mirko Wächter, and Tamim Asfour. Learning Object-Action relations from bimanual human demonstration using graph networks. *IEEE Robotics and Automation Letters*, 5(1):187–194, January 2020. 3.1, 4.1, 3, 4.3.3, 4.4, 5

[5] Yazan Abu Farha and Jurgen Gall. MS-TCN: Multi-Stage temporal convolutional network for action segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3579. IEEE, June 2019. 2.2, 4.2

[6] Pallabi Ghosh, Yi Yao, Larry Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 576–585, 2020. 1, 2.1, 2

[7] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, June 2018. 2.3

[8] Abhinav Gupta, Aniruddha Kembhavi, and Larry S Davis. Observing human-object interactions: using spatial and functional compatibility for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(10):1775–1789, October 2009. 1

[9] Minh Hoai, Zhen-Zhong Lan, and Fernando De la Torre. Joint segmentation and classification of human actions in video. In *CVPR 2011*, pages 3265–3272, June 2011. 2.2

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, November 1997. 2.3

[11] Ninghang Hu, Gwenn Englebienne, Zhongyu Lou, and Ben J A Kröse. Learning latent structure for activity recognition. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1048–1053, May 2014. 2

[12] Ashesh Jain, Amir Roshan Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-RNN: Deep learning on Spatio-Temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317. IEEE, June 2016. 2.1, 2

[13] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017*, November 2016. 3.3

[14] Yun Jiang and Ashutosh Saxena. Modeling High-Dimensional humans for activity anticipation using gaussian process latent CRFs. In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, July 2014. 1, 2.1

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, December 2014. 3.6

[16] Lingpeng Kong, Chris Dyer, and Noah A Smith. Segmental recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, November 2015. 2.2

[17] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from RGB-D videos. *The International journal of robotics research*, 32(8):951–970, July 2013. 1, 2.1, 4.1, 2

[18] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, January 2016. 1, 2.1, 3.1, 4.3.1, 1, 2

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, May 2017. 3.6

[20] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, July 2017. 2.2, 4.2

[21] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal CNNs for Fine-Grained action segmentation. In *Computer Vision – ECCV 2016*, pages 36–52. Springer International Publishing, 2016. 2.2

[22] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6742–6751, June 2018. 2.2

[23] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017*, November 2016. 3.3

[24] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-Based asynchronous sparse convolutional networks. In *Computer Vision – ECCV 2020*, pages 415–431. Springer International Publishing, 2020. 2.3

[25] Romero Morais, Vuong Le, Truyen Tran, and Svetha Venkatesh. Learning to abstract and predict human actions. In *Proceedings of the British Machine Vision Conference 2020*. British Machine Vision Association, September 2020. 4.2

[26] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, pages 3882–3890, 2016. 2.3

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision – ECCV 2016*, pages 483–499. Springer International Publishing, September 2016. 2.1

[28] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. Predicting human activities using stochastic grammar. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1173–1181, October 2017. 2.1, 4.3.1

[29] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4171–4179, Stockholmsmässan, Stockholm Sweden, 2018. PMLR. 2.1, 4.3.1

[30] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning Human-Object interactions by graph parsing neural networks. In *Computer Vision – ECCV 2018*, pages 407–423. Springer International Publishing, 2018. 1, 2.1, 3.1, 2

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, June 2017. 3.6

[32] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. EventNet: Asynchronous recursive event processing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3882–3891, June 2019. 2.3

[33] Ozan Sener and Ashutosh Saxena. rCRF: Recursive belief estimation over CRFs in RGB-D activity videos. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, July 2015. 2.1, 4.3.1, 1

[34] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Intra- and Inter-Action understanding via temporal action parsing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 730–739. IEEE, June 2020. 2.2

[35] Qinfeng Shi, Li Cheng, Li Wang, and Alex Smola. Human action segmentation and recognition using discriminative Semi-Markov models. *International journal of computer vision*, 93(1):22–32, May 2011. 2.2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł Ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017. 3.3

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018*, February 2018. 3.3