Multi-Label Action Anticipation for Real-World Videos with Scene Understanding

Yuqi Zhang, Xiucheng Li, Hao Xie, Weijun Zhuang, Shihui Guo, Zhijun Li

***Abstract*—With human action anticipation becoming an essential tool for many practical applications, there has been an increasing trend in developing more accurate anticipation models in recent years. Most of the existing methods target standard action anticipation datasets, in which they could produce promising results by learning action-level contextual patterns. However, the over-simplified scenarios of standard datasets often do not hold in reality, which hinders them from being applied to real-world applications. To address this, we propose a scene-graph-based novel model SEAD that learns the action anticipation at the high semantic level rather than focusing on the action level. The proposed model is composed of two main modules, 1) the scene prediction module, which predicts future scene graphs using a grammar dictionary, and 2) the action anticipation module, which is responsible for predicting future actions with an LSTM network by taking as input the observed and predicted scene graphs. We evaluate our model on two real-world video datasets (Charades and Home Action Genome) as well as a standard action anticipation dataset (CAD-120) to verify its efficacy. The experimental results show that SEAD is able to outperform existing methods by large margins on the two real-world datasets and can also yield stable predictions on the standard dataset at the same time. In particular, our proposed model surpasses the state-of-the-art methods with mean average precision improvements consistently higher than 65% on the Charades dataset and an average improvement of 40.6% on the Home Action Genome dataset.**

***Index Terms*—Action anticipation, real-world datasets, scene graph, stochastic grammar.**

I. Introduction

HUMAN action anticipation has recently gained increasing attention in both academia and industrial research labs. It finds widespread applications in a variety of practical scenarios such as human-robot interaction, autonomous driving, assistive robotics, video surveillance, and anomaly alert system. For example, the accurate prediction of upcoming human actions enables household robots to offer timely support for their masters; the reliable prediction of pedestrian intentions is crucial for achieving real automated driving safety [1], [2]; it allows the anomaly alert system to trigger corresponding signals if the predicted actions deviate from the correct action sequences so as to avoid accidents.

In light of its great practical values, many human action anticipation methods have been proposed in the past years [3]–[10]. These methods are mostly targeted to the standard datasets—such as 50Salads [11], Breakfast [12], Epic-Kitchens [13], CAD-120 [14], MPII-Cooking [15] and Watch-n-patch [16]—on which they have achieved promising prediction results. The standard datasets mostly are around food preparation and are collected by several different users performing a sequence of fixed actions within the same scene. As a consequence, these standard datasets share three common characteristics: 1) each involved activity is composed of a sequence of fixed actions (the left of Fig. 1 shows the activity *making_cereal* in the CAD-120 dataset, four different users are performing sequences of actions that are almost identical) and the action dependencies are also relatively simple; 2) the scenes remain invariant to the activity genres (in the left of Fig. 1, the activity is repeated by different users in the same scene) and there is almost no complex interaction between human and scene objects (in the left of Fig. 1, the users are only interacting with the bowl); 3) only a single action is occurring at each timestamp. Thus, it is sufficient for the existing methods [3]–[10] that only focus on learning the contextual patterns of action sequences to acquire desirable prediction ability.

However, such three characteristics are tied to the over-simplified cases and do not generalize to the real-world

Fig. 1. An example of the visual comparison between the standard dataset (CAD-120) and the real-world dataset (Charades).

Corresponding authors: Xiucheng Li and Zhijun Li.

Yuqi Zhang and Zhijun Li are with Harbin Institute of Technology, China (e-mails: zhangyuqi2020@gmail.com; lizhijun_os@hit.edu.cn).

Xiucheng Li, Hao Xie, and Weijun Zhuang are with Harbin Institute of Technology, Shenzhen, China (e-mails: lixiucheng@hit.edu.cn; 20s151141@stu.hit.edu.cn; 20S051020@stu.hit.edu.cn).

Shihui Guo is with Xiamen University, China (e-mail: guoshihui@xmu.edu.cn).

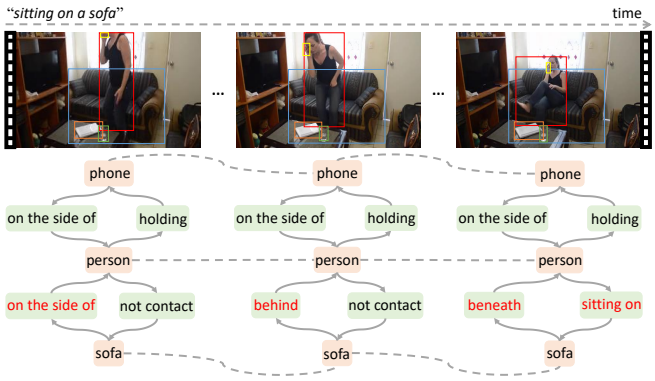


Fig. 2. Key frames of action *sitting on a sofa* and their corresponding scene graphs consisting of interaction tuples such as $\langle \text{sofa}, \text{on the side of}, \text{person} \rangle$.

datasets, e.g., Charades [17], which are more realistic and collected from our real-life scenarios. In contrast, the same activity involved in real-world datasets may have diverse variants, and the action dependency patterns could also be complex, as shown in the right of Fig. 1, various actions (*holding a dish*, *pouring*, *awakening*, etc.) can precede action *sitting on a sofa*. The scenes in which the actions occur exhibit great variability (as shown in the right of Fig. 1, the same action *sitting on a sofa* could occur under completely distinct scenes) and there may be very complex interactions between humans and scene objects. Moreover, multiple actions can occur simultaneously and tangle with each other, e.g., *sitting on a sofa* and *drinking from a glass* are co-occurring in the right of Fig. 1, and thus the real-world datasets are usually multi-labeled. Since the existing human action anticipation approaches are developed specifically for the standard datasets, such discrepancies between the simplified and realistic cases make it very challenging to apply them to real-world datasets. In particular, the diverse variants of activities, complex action dependency patterns, and variability of scenes would incur significant extra learning burdens for the existing methods focusing on learning sequential patterns at the action level, in the meanwhile, they also fail to explicitly utilize the complex interactions between humans and objects; moreover, they are even inapplicable to the cases in which multiple actions are co-occurring. Consequently, the existing methods still suffer and yield poor performance on real-world datasets.

To address these challenges, we propose to learn the action anticipation at a high semantic level rather than focusing on the low action level. Our method draws inspiration from the fact [18] that an action can be regarded as the interactions between humans and objects, denoted by tuples $\langle \text{person}, \text{relationship}, \text{object} \rangle$. Fig. 2 illustrates an example, in which $\langle \text{person}, \text{sitting on}, \text{sofa} \rangle$, etc., are the interaction tuples. Since we characterize each action by using the high-level semantic representation, interaction tuple, which explicitly reveals the human intention and thus it would be much easier to anticipate the upcoming actions with such intention-aware representation than directly manipulating the low-level action features. For instance, regarding the action *sitting on a sofa* in Fig. 2, given the change of the interaction tuple from $\langle \text{sofa}, \text{on the side of}, \text{person} \rangle$ to $\langle \text{sofa}, \text{behind}, \text{person} \rangle$, we can confidently predict that the person is about to sit on the sofa without having to acquire all its preceding actions (*talking on a phone*, *holding a phone*, etc.). By contrast, to make accurate predictions, the existing methods [3]–[10] have to recognize and memorize the complex action dependencies underlying the long action sequences.

To this end, we develop a Scene and Action Joint Prediction (SEAD) model to boost the real-world human action anticipation, based on scene graphs comprising of a collection of human-object interaction tuples [19]. The key idea is to represent video frames with scene graphs and then jointly forecast the future scene graphs and human actions by using the obtained scene graphs up to the present moment, which stands in contrast to the existing methods that directly operate on the action level. More specifically, our proposed model SEAD consists of two main modules. The first one is a scene prediction module, which aims to predict the future scene graphs by using a grammar dictionary; the second action anticipation module is responsible for predicting future human actions with an LSTM network by taking as input the observed and predicted scene graphs. Herein, we use the scene graphs to capture both humans and objects as well as their relationships. The benefits are threefold: 1) modeling the human-object interactions explicitly makes SEAD less insensitive to the diverse variants of activities (action sequences) as well as various scenes that are irrelevant to the human actions (e.g., background); 2) the human-object interaction representations are more intention-aware, thus it relieves SEAD from the burden of identifying and memorizing the long and complex dependencies in the action sequences, especially for the inter-dependencies between activities; 3) the scene graphs enable us to capture multiple human-object interactions simultaneously, therefore SEAD is able to handle the multi-label action anticipation in a natural fashion.

In summary, our contributions are as follows. 1) To the best of our knowledge, the scene graphs are first introduced to address the human action anticipation tasks. Our proposed model SEAD approaches the action anticipation from the scene graph view rather than focusing on a low action level. It is capable of handling diverse action sequences, various scenes, complex human-object interactions, and co-occurring actions. 2) We propose to predict the scene graphs and upcoming actions simultaneously, and the predicted scene graphs are used to aid action anticipation which enables more accurate predicting results. 3) We conduct extensive experiments on two real-world video datasets (Charades and Home Action Genome [20]) and a standard action anticipation dataset (CAD-120). The results show that our proposed model SEAD outperforms the existing methods by large margins on two real-world datasets and can produce stable predictions on the standard dataset. In particular, our proposed model surpasses the state-of-the-art methods with mean average precision improvements consistently higher than 65% on the Charades dataset and an average improvement of 40.6% on the Home Action Genome dataset.

The rest of the paper is organized as follows. We first review the related work of human action anticipation and scene graph

in Section II, and then present the background knowledge on stochastic grammar [21] and its parsing algorithm Earley parser [22] in Section III. The proposed model is detailed in Section IV, and Section V presents experimental results. Finally, we conclude the manuscript with Section VI.

II. Related Work

A. Human Action Anticipation

The widespread applications of human action anticipation have spurred many proposals being developed in the literature. In the early stage, due to their simplicity and interpretability, the researchers in the community often rely on the grammar tools [3], [23], [24] to recognize the patterns of activities consisting of fixed action sequences. Holtzen *et al.* [23] propose to predict human intention by using the grammar model to characterize the relationships between actions and intentions. Xiong *et al.* [24] develop an And-Or Graph-based grammar model to predict future actions for robot planning. To further enhance their prediction ability, Qi *et al.* [3] propose a spatial-temporal And-Or Graph model and verify its efficacy on the standard CAD-120 dataset. However, these grammar-based methods are only able to handle the action sequences with clear compositional structures and require extra manual feature extraction steps.

To eliminate the reliance on manual feature engineering, many deep neural network-based methods are proposed for human action anticipation to harness their automatic representation learning capability. In particular, Recurrent Neural Nets (RNNs) and their variants are widely adopted for distilling informative features from action frames and learning the sequential patterns underlying the action sequences. Abu Farha *et al.* [25] propose two methods, CNN-based and RNN-based, to predict future actions and their duration respectively. Furnari *et al.* [26] learn action anticipation from first-person videos using two LSTMs to summarize the previous and predict the future. Ng *et al.* [6] develop a weakly supervised model to forecast future action sequences by using a GRU-based encoder-decoder architecture. Recently, there has been a trend to combine grammar methods and deep neural networks for action anticipation so as to take advantage of both worlds. Qi *et al.* [5] generalize the Earley parser and integrate it with deep neural networks to process unsegmented and unlabeled sequential data. Piergiovanni *et al.* [8] propose an Adversarial Generative Grammar model for future human actions prediction. These deep neural network-based models and hybrid methods deliver decent performance on standard datasets. However, since all of them focus on learning sequential patterns at the action level, they are sensitive to the diverse variants of action sequences and variability of scenes, and the complex action dependency patterns will also impose significant learning burdens on them. Not surprisingly, they perform poorly on real-world datasets.

In recent years, several researchers have turned to learning fine-grained representations for action anticipation. Mahmud *et al.* [27] propose to train the LSTM network by using both motion features and object features for future prediction. Roy *et al.* [28] develop a multi-modal transformer that jointly utilizes the human-object, motion, and spatiotemporal representations to anticipate the upcoming actions. The fine-grained representations enable these models to achieve much better results on the standard datasets. However, as they fail to consider the rich interactions between humans and objects, these methods are not able to handle the co-occurring actions well and their performance on real-world datasets is still less desirable. Nonetheless, the empirical success of fine-grained representations proves their usefulness and it inspires us to introduce scene graphs to learn more fine-grained human-object interaction representations for this problem.

B. Scene Graph

The scene graph is a sort of structural representation of the image in the form of a graph. The graph nodes denote the objects in the image and the edges correspond to the pairwise relationships between objects. It is a high-level semantic representation of the image and could facilitate a wide spectrum of computer vision tasks (as we will discuss later). Hence, it has attracted a lot of attention [29]–[31] in the community since its emergence. The proposal [29] develops the first end-to-end scene graph generation model, it generates the scene graph by iteratively refining the predicted results of the RNNs through message passing. As the number of edges in the graph grows quadratically with the number of nodes, such a progressive generation manner is often very costly. To speed up the scene graph generation, Li *et al.* [30] propose to factorize the whole graph into a collection of subgraphs by using a bottom-up clustering method; Yang *et al.* [31] propose the relation proposal network to eliminate unnecessary computation by pruning edges that correspond to unlikely relations. Very recently, several new directions on scene graphs have been studied [32]–[34]. The proposal [32] studies the scene graph generation in a semi-supervised way, whereas the proposal [33] investigates the problem of unbiased scene graph generation. Cong *et al.* [34] study the problem of dynamic scene graph generation for videos, they propose a Spatial-Temporal Transformer (STTran) by exploring both the spatial correlation and temporal dependency underlying the consecutive frames. STTran is able to produce scene graphs in a dynamic manner, which could be used as more fine-grained frame representations for action anticipation tasks. Thus, it is employed as the workhorse of our proposed model.

Due to its great abstract semantic representation ability, the scene graph has been explored and proves useful in boosting a wide variety of image processing and computer vision tasks such as image captioning [35], [36], image retrieval [37], visual question answering [38], [39], image generation [40], [41], and action recognition [18], [20]. In particular, the proposals [18], [20] have demonstrated the effectiveness of scene graphs in action recognition, the recognition accuracy gains considerable improvement with the aid of this high-level semantic representation on the real-world datasets. However, the scene graph has been little investigated in action anticipation and it remains open on how to effectively apply such powerful representation to this problem. To the best of our knowledge, this is the first work that explores its powerful representation capability in action anticipation.

III. Preliminaries

A. Stochastic Grammar

Context-free grammar (CFG) is a type of formal grammar, which contains a set of rules describing all possible sentences in a formal language [21]. Formally, a CFG in Chomsky Normal Form is defined by a 4-tuple $\phi = (S, N, T, R)$ where S is the start symbol of the language; T is a finite set of terminals representing the words in the language and cannot be further expanded; N is the set of symbols that can be replaced by a sequence of terminal or nonterminal symbols; the production rules R specify the manner in which the terminals and nonterminals can be combined to form strings or sentences. The production rules R are represented in the following form:

$$A \rightarrow aB \mid b, \tag{1}$$

where A and B are non-terminals in N, a, b are terminals in T. It means the nonterminal symbol A can be replaced by either expression aB or b on the right-hand side.

Since there is often a certain probability of the occurrence of signals in the real world, stochastic context-free grammar (SCFG) associates each production rule with a probability [21]. The SCFG can be formally defined by a 5-tuple $\phi = (S, N, T, R, P)$ where P is a set of probabilities on production rules R. In this paper, we use stochastic grammar to characterize the patterns of interactions between humans and each particular object when constructing the grammar dictionary.

B. Earley Parser

The Earley parser [22], an efficient grammar parsing algorithm, is used to predict the future interactions between humans and each object in this paper. To describe the Earley parser, α, β, and γ represent any terminal or nonterminal string. A and B are single nonterminal symbols, and a is a terminal symbol. The dot in the production rule of $A \rightarrow \alpha \cdot \beta$ indicates that α has been parsed, and β is to be expected.

The input position k represents the position after the k-th token is accepted. At each k, the parser will generate a state set $L(k)$ in which each state is a tuple $(A \rightarrow \alpha \cdot \beta, n)$ consisting of:

- $A \rightarrow \alpha \cdot \beta$: the currently being matched production rule.
- n: the position n in the input where the matching of the production rule began.

Seeded with $L(0)$ that only contains the top-level rule, then the parser repeatedly executes the following three basic operations:

- **Prediction**: for every state of the form $(A \rightarrow \alpha \cdot B\beta, n)$ in $L(k)$, find the production rule with B on the left-hand side (e.g., $B \rightarrow \gamma$) in the grammar and add $(B \rightarrow \cdot\gamma, k)$ to $L(k)$.
- **Scanning**: for every state of the form $(A \rightarrow \alpha \cdot a\beta, n)$ in $L(k)$, add $(A \rightarrow \alpha a \cdot \beta, n)$ to $L(k{+}1)$.
- **Completion**: for every state of the form $(A \rightarrow \gamma\cdot, m)$ in $L(k)$, find the state in the form of $(B \rightarrow \alpha \cdot A\beta, n)$ in $L(m)$ and add $(B \rightarrow \alpha A \cdot \beta, n)$ to $L(k)$.

The parser repeatedly performs these three operations until no new states can be added to the state set.

IV. Scene and Action Joint Prediction Model

Fig. 3 presents the overall framework of our proposed method SEAD, which consists of two main modules, the scene prediction module and action anticipation module, presented in Section IV-A and IV-B, respectively. The scene prediction module intends to predict the future scene graphs by using the grammar dictionary, whereas the action anticipation module is used to anticipate the upcoming actions with an LSTM network by taking as input the observed and predicted scene graphs. We now illustrate these two modules in detail.

A. Scene Prediction Module

The state-of-the-art action anticipation methods learn to predict the future by identifying the dependency patterns of action sequences and directly manipulating the raw frames. As a consequence, these methods are usually sensitive to the variation of action sequences and scene changes (e.g., background) and fail to utilize the rich interactions between humans and objects. To address these limitations, we instead propose to learn the prediction model with a high-level semantic representation—scene graph.

Scene Graph Representation. The scene graph $G = (O, R)$ of an image consists of a collection of objects $O = \{o_i \mid 1 \leq i \leq K\}$ and the relationship set $R = \{r_{ij} \mid 1 \leq i \leq K, 1 \leq j \leq n_i\}$, where K is the number of objects, r_{ij} indicates the j-th relationship between human and object i (o_i), and n_i is the total number of relationships associated with o_i.

Hence, the scene graphs of successive video frames can also be characterized as multiple sequences of human-object interaction tuple pair, i.e., ⟨person, relationship, object⟩ and ⟨object, relationship, person⟩. For example, the scene graphs in Fig. 2 can be represented as the sequence of human-phone interaction tuple pair and the sequence of human-sofa interaction tuple pair (e.g., ⟨person, not contact, sofa⟩ and ⟨sofa, on the side of, person⟩, ⟨person, not contact, sofa⟩ and ⟨sofa, behind, person⟩, ⟨person, sitting on, sofa⟩ and ⟨sofa, beneath, person⟩ is a human-sofa interaction tuple pair sequence). Such human-object interaction tuple pair sequences are much more semantically meaningful and intention-aware. As a result, it relieves the model from the burden of identifying and memorizing the long and complex dependency patterns underlying action sequences. It is also more robust to the variation of action sequences and scene variability, and moreover, it enables us to model the interactions between humans and objects very naturally. In addition, the intention-aware tuple pair sequences between humans and each particular object have more clear compositional structures and thus it permits us to use stochastic grammar to capture the dynamics of human-object interaction, which will result in a more lightweight and interpretable forecasting model.

There are generally three types of human-object relationships existing in the scene graph: spatial relationship, contact relationship, and attention relationship. The spatial relationship specifies the spatial layout of objects, e.g., "on the side of"; the contact relationship describes the physical contact between human and object, e.g., "drinking from"; whereas the attention relationship indicates where the human is paying attention to,

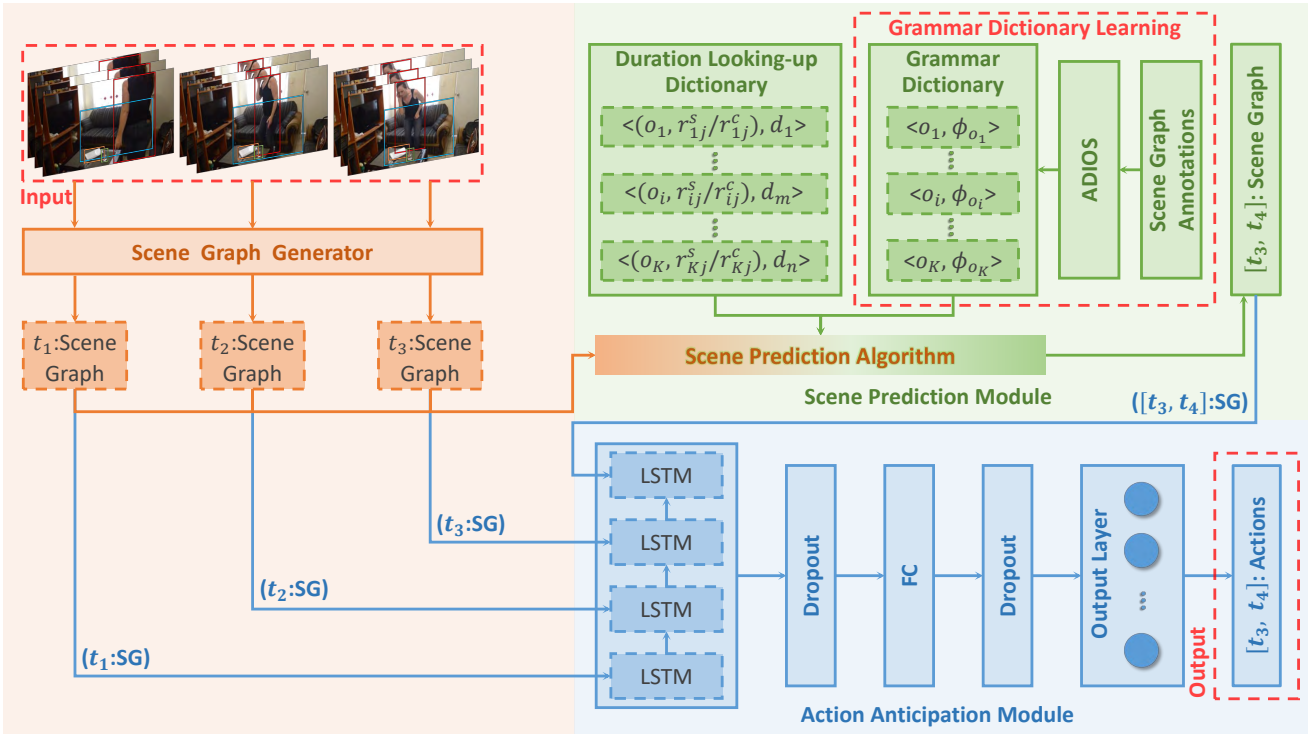


Fig. 3. The overall framework of our proposed method SEAD mainly consists of a scene prediction module and an action anticipation module. Given the observed video clip spanning from t_1 to t_3, the scene graph generator first maps observed frames into scene graph sequence, which is fed into the scene prediction module to predict the scene graphs within the period $[t_3, t_4]$ (there is only one predicted scene graph in this figure). This prediction is accomplished by employing the Scene Prediction Algorithm 1, which takes as inputs $\mathcal{D}$, $\mathcal{T}$, the observed SG sequence, and $\Delta t = t_4 - t_3$. Finally, the action anticipation module predicts the actions within the period $[t_3, t_4]$ with an LSTM network by taking as the input the observed and predicted scene graphs.

e.g., "looking at". In this paper, we mainly explore the usage of the first two types of relationships, that is, spatial relationship and contact relationship, since they are more informative to the upcoming actions as explained as follows.

A lot of evidence in cognitive science already shows that the actions in the human brain can generally be divided into two classes based on the scale, namely, B-action and H-action. The B-action is relevant to the whole body activity [42] such as sitting, standing, and sleeping, whereas the H-action is linked with the hand movement such as holding and touching. Correspondingly, psychological research also finds that objects can be categorized into two types, B-object and H-object. The B-object (resp. H-object) corresponds to B-action (resp. H-action) and it includes the objects relevant to the entire body activity (resp. the hand movement), e.g., furniture (resp. the glass on the table). Intuitively, 1) a change in the spatial relationship involving B-object usually implies the occurrence of B-action, e.g., the change from $\langle$sofa, on the side of, person$\rangle$ to $\langle$sofa, behind, person$\rangle$ often means the intention of *sitting on a sofa*; 2) a change in the contact relationship involving H-object usually indicates the occurrence of H-action, e.g., the change from $\langle$person, not contact, glass$\rangle$ to $\langle$person, touching, glass$\rangle$ often means that the person is very likely to perform the action of *holding a glass*.

Grammar Dictionary Learning. Motivated by this intuition, we propose to learn the stochastic grammar ϕ_{o_i} for each $o_i \in O$. The concept of a grammar dictionary refers to a dictionary that stores the stochastic grammars associated with all objects, that is, the keys encompass all objects in O, and the corresponding values represent their respective stochastic grammars, as exemplified in Equation 3. Notably, for the human-object interaction tuple pair $\langle$person, relationship, object$\rangle$ and $\langle$object, relationship, person$\rangle$, the relationship of tuple $\langle$person, relationship, object$\rangle$ is the contact relationship (denoted by r^c), and the relationship of tuple $\langle$object, relationship, person$\rangle$ is the spatial relationship (denoted by r^s), there is a one-to-one correspondence between spatial and contact relationships for a given object in the scene graph. Hence, to obtain ϕ_{o_i}, for each training video l with its scene graph annotations containing o_i, we first extract r^s and r^c of o_i from the frames with scene graph annotations, as well as reindex and deduplicate them by their chronological order to form a sequence of relationship pair $S_{il} = \{r^s_{i1}/r^c_{i1}, r^s_{i2}/r^c_{i2}, \ldots\}$. We perform this operation on all training videos whose scene graph annotations contain o_i and get their corresponding relationship pair sequences to yield $\mathcal{C}_i$,

$$\mathcal{C}_i = \{S_{il} \mid 1 \leq l \leq n_l\}, \tag{2}$$

which is referred to as the corpus of o_i, and n_l is the total number of training videos whose scene graph annotations contain o_i. Then for each o_i we learn its stochastic grammar ϕ_{o_i} on its corpus $\mathcal{C}_i$ and then use ϕ_{o_i} to construct the grammar dictionary as:

$$\mathcal{D} = \{o_i : \phi_{o_i} \mid 1 \leq i \leq K\}, \tag{3}$$

where o_i, ϕ_{o_i} are the keys and values of the dictionary,

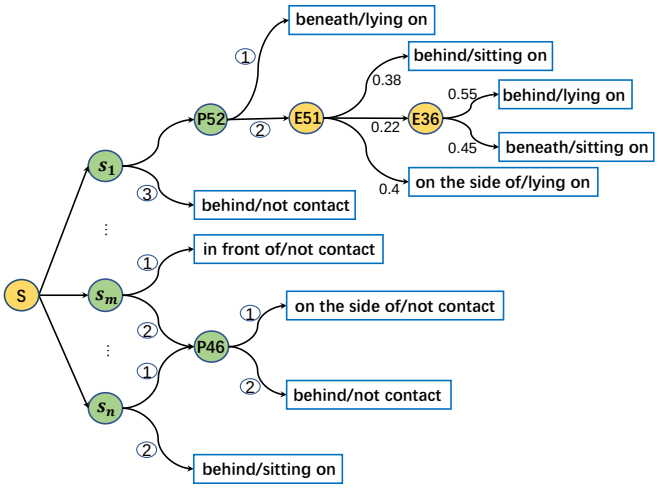


Fig. 4. Some examples of the stochastic grammar of object sofa. The yellow nodes represent root and equivalence classes (Or nodes). The green nodes represent significant patterns (And nodes).

respectively. This grammar dictionary plays a central role in our subsequent scene prediction task, as will be discussed later.

In this paper, we opt for ADIOS algorithm [43] to learn the stochastic grammar. ADIOS is a widely adopted unsupervised grammar induction algorithm for hierarchical structure learning. The algorithm takes a set of sequences as input and produces the hidden compositional structure of these sequences. In our case, given an object o_i, the algorithm starts with loading its corpus $\mathcal{C}_i$ to construct a directed graph, whose vertices are $r^s_{ij}/r^c_{ij} \in \mathcal{C}_i$ with two augmented special vertices, *begin* and *end*. Each sequence in the corpus defines a separate path on the graph, starting at *begin* and ending at *end*. The algorithm then iteratively adds two types of vertices—the significant pattern vertices and equivalence class vertices—into the graph to form the hierarchical structure. The iteration is repeated until no new significant pattern is found. The output of the algorithm is referred to as And-Or Graph (AOG), which represents the learned hidden hierarchical structure. In AOG, there are two kinds of nodes, terminal nodes corresponding to the elements of $\mathcal{C}_i$ and nonterminal nodes representing the grammar rules. The nonterminal nodes can be further divided into And nodes and Or nodes. The And nodes, converted from the significant patterns, represent the composition and chronological order; Or nodes, converted from the equivalence classes, define the probability of child node selection. Fig. 4 shows an instance of And-Or Graph. The blue color boxes in the graph are the terminal nodes; $P46$ (significant pattern) is a And node with two child nodes–on the side of/not contact and behind/not contact, where the number on the edge indicates the temporal order of the corresponding child node; whereas $E36$ (equivalence class) is an Or node with child nodes–behind/lying on and beneath/sitting on, the numbers on edges indicate the probabilities (0.55 vs 0.45) of selecting the corresponding children when traversing the graph.

Grammar Dictionary-Based Scene Prediction. We now elaborate on how to use the learned grammar dictionary $\mathcal{D}$ to predict the scene graphs within the next Δt seconds.

Algorithm 1: Scene Prediction Algorithm

Input: $\mathcal{D}$ (grammar dictionary), $\mathcal{T}$ (duration dictionary), O_S (observed scene graphs), Δt
Output: P_S (predicted scene graphs over the next Δt seconds)

$O_S \leftarrow$ `RemoveConsecutive` (O_S);
$\mathcal{S} \leftarrow$ Initialize an empty dictionary;
for $G \in O_S$ **do**
  for $o_i \in G$ **do**
    Extract the spatial-contact event r^s_{ij}/r^c_{ij} of o_i;
    if $o_i \notin \mathcal{S}$ **then**
      $\mathcal{S}[o_i] \leftarrow r^s_{ij}/r^c_{ij}$;
    else
      $\mathcal{S}[o_i] \leftarrow$ `Append` $(\mathcal{S}[o_i], r^s_{ij}/r^c_{ij})$;
for $o \in \mathcal{S}$ **do**
  $s \leftarrow$ `RemoveConsecutive` $(\mathcal{S}[o])$;
  $\phi_o \leftarrow \mathcal{D}[o]$;
  $s_n \leftarrow$ Get the last element of s;
  if $(o, s_n) \notin \mathcal{T}$ **then**
    $s_n \leftarrow$ `FindClosest` (s_n);
    $t_c \leftarrow \mathcal{T}[(o, s_n)]$;
  else
    $t_c \leftarrow \mathcal{T}[(o, s_n)]$;
  while $t_c < \Delta t$ **do**
    $e \leftarrow$ `EarleyParser` (ϕ_o, s);
    $t_c \leftarrow t_c + \mathcal{T}[(o, e)]$;
    $s \leftarrow$ `Append` (s, e);
    $P_S \leftarrow$ `Append` (P_S, e);
return P_S;

In our following discussion, we will refer to the spatial-contact relationship pair r^s/r^c as the spatial-contact event. Note that each spatial-contact event has a particular value (e.g., beneath/sitting on) as well as a duration indicating how long the event lasts.

Recall that the predicted scene graphs only serve as the intermediate representations to assist the subsequent action anticipation and we are only interested in the spatial and contact relationships in the scene graphs, thus it suffices to predict the spatial-contact event sequence for each object within the next Δt seconds. To achieve this, our general idea is, for each object, to employ the Earley parser to iteratively predict the next most possible event and collect the predicted event sequence along the iteration, and this process is terminated when the sum of collected event duration reaches Δt. But how could we know the duration of each spatial-contact event? It is very natural to adopt the average duration (computed from the training datasets) of each particular event since different events may have different duration, indeed, as Fig. 5-(b) shows, for a given object chair, the average duration varies against events. However, we also empirically find that the duration may also change against objects even for the same event. This is shown in Fig .5-(a), given the event in front of/holding, the duration

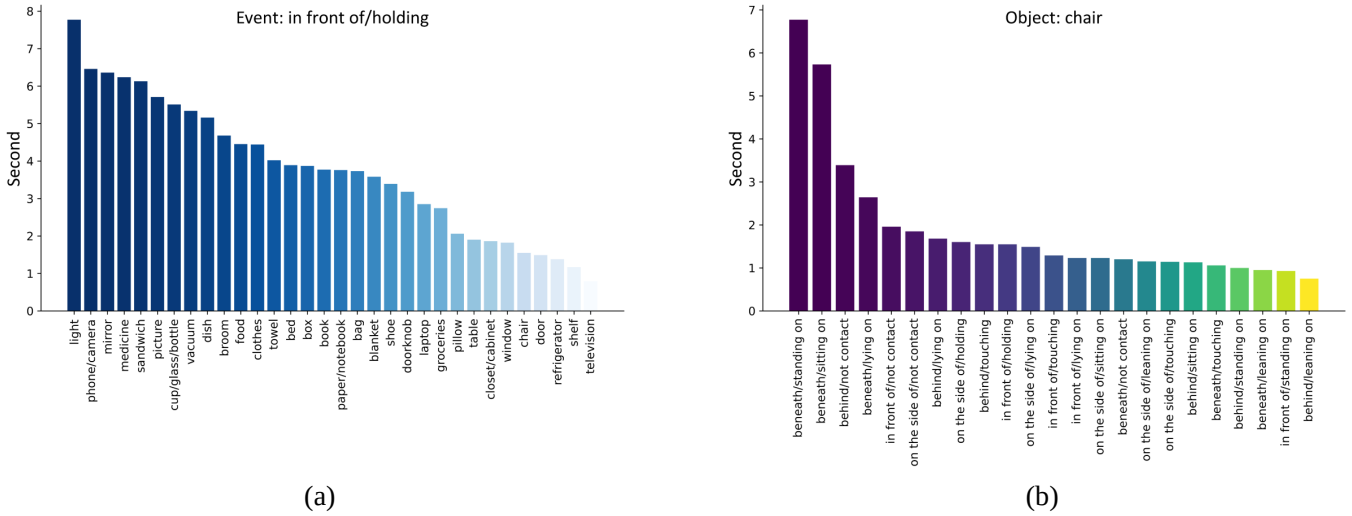


Fig. 5. (a) indicates the average duration of the event in front of/holding regarding different objects. (b) indicates the average duration of different spatial-contact events about the object chair.

of different objects varies significantly. In other words, the duration of the event depends not only on the event value (itself) but also on the involved object. For this reason, we construct a duration looking-up dictionary $\mathcal{T}$ as follows,

$$\mathcal{T} = \left\{(o_i, r^s_{ij}/r^c_{ij}) : d \mid 1 \leq i \leq K, 1 \leq j \leq |\mathcal{C}_i|\right\}, \quad (4)$$

in which the keys $(o_i, r^s_{ij}/r^c_{ij})$ are the (object, event) pairs, values d are the corresponding average duration of the events r^s_{ij}/r^c_{ij} regarding the objects o_i, and $|\mathcal{C}_i|$ denotes the number of distinct spatial-contact events in $\mathcal{C}_i$. In light of this observation, we develop the Scene Prediction Algorithm 1, which is illustrated as follows.

The algorithm takes as input the learned grammar dictionary $\mathcal{D}$, duration looking-up dictionary $\mathcal{T}$ as well as the observed scene graphs O_S, and it returns the predicted scene graphs P_S. We first remove the consecutive duplicated scene graphs in the 1_{th} line of the Algorithm 1. Then from lines 3 to 9, we extract the spatial-contact event sequence from O_S for each object o_i and store the sequence into a temporal dictionary $\mathcal{S}$ with o_i as keys. From lines 10 to 23, for each object o, we run the Earley parser to iteratively predict the next most likely spatial-contact event e corresponding to o. Since the Earley parser requires the input sequence to have no consecutive duplicated symbols, we ensure this by invoking function `RemoveConsecutive` on the event sequence $\mathcal{S}[o]$. The Earley parser predicts the next event e by taking as input the learned grammar ϕ_o and spatial-contact event sequence s in line 20. More specifically, the Earley parser initially parses ϕ_o to generate a set of potential future spatial-contact events following s by employing the three operations introduced in Section III-B. Subsequently, the probability values in ϕ_o are leveraged to calculate the likelihood associated with these potential events to predict the next event e with the highest probability. The line 21 accumulates the expected duration of the predicted spatial-contact event e, which is retrieved from the duration looking-up dictionary $\mathcal{T}$ by using the (object, event) tuple as key. The newly predicted event is also being appended to event sequence s and collected to form the scene graphs in lines 22 and 23, respectively. We complete the generation of a spatial-contact event sequence for a given object when the accumulated duration reaches or exceeds Δt in line 19.

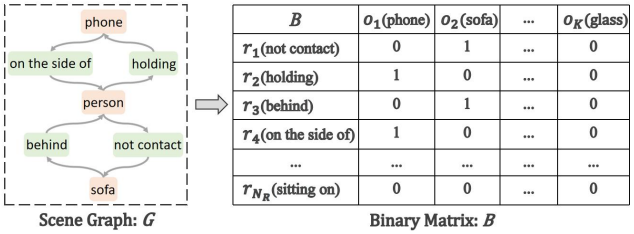
B. Action Anticipation Module

In this section, we illustrate how to predict future actions with the aid of scene graph representations. The idea is to learn a sequential forecasting model by taking as input the scene graph sequence up to the predicted timestamp. Suppose the present moment is t_0, in the training stage, we can access and treat as input all observations up to the future timestamp $t_0 + \Delta t$ whereas in the test phase only the observations up to present moment t_0 are available, thus we first predict the scene graphs between t_0 and $t_0 + \Delta t$ and then take as input both the observed and predicted scene graphs. In this paper, we choose LSTM as the forecasting function since it is widely adopted for sequential learning in practice.

Scene Graph Encoding. To feed the scene graph $G = (O, R)$ into the LSTM network, we propose to encode the G into a binary matrix B of size $K \times N_R$ where N_R is the number of distinct relationships in total. Let $\mathrm{ord}(r_{ij})$ denotes the order of relationship r_{ij}, that is, $1 \leq \mathrm{ord}(r_{ij}) \leq N_R$, then we set the corresponding entry $B_{i,\mathrm{ord}(r_{ij})}$ to 1 if the relationship r_{ij} is involved in the interaction of the person and object o_i in the given graph G and 0 otherwise. More formally,

$$B_{i,\mathrm{ord}(r_{ij})} = \begin{cases} 1, & \text{if } o_i \in O \text{ and } r_{ij} \in R, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Note that given the order of relationships (rows) and objects (columns), the generated binary matrix B will be unique to each scene graph G, and we use a fixed order in our implementation to ensure consistency. Fig. 6 illustrates an example of encoding a scene graph G with a binary matrix B, in which, for the spatial-contact event on the side of/holding



B	o_1(phone)	o_2(sofa)	...	o_K(glass)
r_1(not contact)	0	1	...	0
r_2(holding)	1	0	...	0
r_3(behind)	0	1	...	0
r_4(on the side of)	1	0	...	0
...	...	...	...	...
r_{N_R}(sitting on)	0	0	...	0

Fig. 6. An example of encoding a scene graph G with a binary matrix B.

(resp. behind/not contact) involving the object phone (resp. sofa), the corresponding entries $B_{1,2}$ and $B_{1,4}$ (resp. $B_{2,1}$ and $B_{2,3}$) are set to 1 in the binary matrix B.

Action Anticipation Model Training. In the real-world action anticipation datasets, there is a collection of video clips. Each video clip X contains multiple human-labeled actions $\{Y_m\}_{m=1}^{M}$ at M timestamps $\{t_m\}_{m=1}^{M}$ (note that M is a variable that depends on the specific video clip X), where Y_m is a binary vector with length equal to the number of distinct actions in the entire dataset and its k-th entry is 1 if k-th action occurs at timestamp t_m and 0 otherwise (Y_m may contain multiple one-value entries if multiple actions occur at timestamp t_m). We build the action anticipation model by establishing the mapping between the video clip in the past T seconds and Y_m. To this end, we check the annotated scene graphs during $[t_m - T, t_m]$ frame by frame and add into $\mathcal{X}_m$ the scene graphs that are different from their last ones, in other words, we remove the consecutive duplicated ones in the time window $[t_m - T, t_m]$. Next, we transform each scene graph in $\mathcal{X}_m$ into its binary matrix encoding by using Equation 5 (as illustrated in Fig. 6). Finally, we flatten the binary matrices into vectors and feed the vector sequence $\mathbf{x}_m$ into LSTM to predict the action labels Y_m. Specifically, we adopt a one-layered LSTM with hidden size 128 and the output of LSTM is passed through a sigmoid function, and the binary cross entropy loss is used to train the model. Let $\mathbf{h}_m = \mathrm{LSTM}(\mathbf{x}_m)$ then the loss is defined as

$$\mathcal{L} = -\frac{1}{n}\sum_{k} Y_{mk}\log(\sigma(\mathbf{h}_m)) + (1 - Y_{mk})\log(1 - \sigma(\mathbf{h}_m)), \tag{6}$$

where the $\sigma(\cdot)$ denotes the sigmoid function. Since different scenes might have different duration, $\mathbf{x}_m$ would be a variable length sequence. Even though the LSTM could handle variable-length sequences in principle, we empirically find that only using the last four elements in the sequence could give rise to a good performance.

Model Prediction. Given the trained action anticipation model, the prediction of SEAD for future actions works as follows. Since the average duration of the spatial-contact events (r_{ij}^s/r_{ij}^c) corresponding to the object o_i mostly takes longer than one second, SEAD processes the scene graph with a sampling rate of one second in the prediction phase. Specifically, given an observed video clip spanning from 0 to t_0, we first adopt the scene graph generator [34] to map every frame into its scene graph (with a sampling rate of one second), so as to generate the observed scene graph sequence O_S. Then we predict the scene graphs P_S within the period $[t_0, t_0 + \Delta t]$ by taking as input the grammar dictionary $\mathcal{D}$, duration looking-up dictionary $\mathcal{T}$, the observed scene graphs O_S, and Δt with the Algorithm 1. Next, we remove the consecutive duplicated ones from both the observed scene graphs O_S and the predicted ones P_S and transform each scene graph into its binary matrix encoding by using Equation 5. Eventually, we flatten the binary matrices into vectors and feed the vector sequence $\mathbf{x}_{t_0+\Delta t}$ into LSTM to predict the actions of all frames during $[t_0 + \Delta t - 1, t_0 + \Delta t]$.

V. Experiments

In this section, we evaluate our proposed method SEAD against the baseline methods on three datasets. We first present the implementation details and evaluation metrics in Section V-A, then we study the performance of different methods on Charades (real-world dataset), Home Action Genome (real-world dataset), and CAD-120 (standard dataset) in Section V-B, Section V-C, and Section V-D, respectively. Our code is available at https://github.com/YuqiZhang2020/SEAD/tree/master.

A. Setting up

Implementation Details. We opt for STTran [34] as the scene graph generator, which detects the objects and predicts relationship labels of object pairs. The FasterRCNN based on ResNet101 serves as the object detector, and success is determined by an overlap of at least 0.5 IoU between the predicted box and the ground-truth box. Meanwhile, the scene graph generation follows the strategy "*With Constraint*" [34] which predicts the most critical relationships between an object pair. For the learning of the grammar dictionary, the parameters of ADIOS algorithm are set as follows: the size of the context window used for searching the equivalence classes is set to 5, and the minimum overlap for bootstrapping equivalence classes is set to 0.65; the parameters that control the definition of pattern significance α [43] and η [43] are set to 0.01 and 0.9, respectively. We adopt Adam [44] with a learning rate of 0.001 as the optimizer and use the Dropout with a dropping probability of 0.2 after each layer. We train the model for 50 epochs with the mini-batch size 72 and the early stopping is used on validation datasets.

Evaluation Metrics. We adopt the mean average precision (mAP) as the main evaluation metric [7], [8], which is widely used to evaluate multi-label classification tasks in the literature. In addition, the accuracy, macro precision (Macro Pre.), macro recall (Macro Rec.), and macro F1-score (Macro F1) [3], [5] are also chosen to evaluate the performance of SEAD. For all these adopted metrics, the higher values indicate better prediction performance.

B. Action Anticipation on Charades

Dataset. Charades is a very challenging video dataset with unstructured daily activities recorded in the indoor environment such as the living room, dining room, bathroom, kitchen, and recreation room. It is a typical multi-labeled real-world dataset, which possesses the characteristics of our real world such as

TABLE I
PREDICTION MAP FOR ACTIONS WITHIN THE NEXT 45 SECONDS ON THE CHARADES DATASET

Methods	1s	2s	5s	10s	20s	30s	45s
RandomPred	2.4	2.4	2.4	2.4	2.4	2.4	2.4
LastPred	15.1	13.8	12.8	10.2	7.6	6.2	5.7
FC Autoregressive	13.5	14.0	12.6	6.7	3.7	3.5	5.1
FC Direct	15.2	14.5	12.2	9.1	6.6	6.5	5.5
LSTM	12.6	12.7	12.4	10.8	7.0	6.1	5.4
Grammar [7]	15.7	14.8	12.9	11.2	8.5	6.6	8.5
AGG [8]	17.0	15.9	13.4	10.7	7.8	7.2	9.8
SEAD	**28.1**	**26.9**	**24.2**	**21.4**	**18.3**	**17.7**	**18.2**
Improvement (%)	65.3	69.2	80.6	91.1	115.3	145.8	85.7

TABLE II
FUTURE 3-SECOND PREDICTION RESULTS REGARDING THE SEQ_LEN

SEQ_LEN	mAP	Macro Precision	Macro Recall	Macro F1
1	20.9	12.6	10.4	9.4
2	24.3	16.2	12.8	11.9
3	25.2	16.9	13.6	12.9
4	**25.3**	17.3	**13.9**	**13.0**
5	25.1	**18.1**	13.3	12.6

TABLE III
FUTURE 3-SECOND PREDICTION RESULTS FOR ABLATION STUDY ON EQUATION 4

$\mathcal{T}$	mAP	Macro Precision	Macro Recall	Macro F1
w/o_o_i	23.8	15.8	**13.9**	12.8
with_o_i	**25.3**	**17.3**	**13.9**	**13.0**

TABLE IV
FUTURE 3-SECOND PREDICTION RESULTS WITH DIFFERENT OBJECT DETECTION METHODS

Methods	mAP	Macro Pre.	Macro Rec.	Macro F1
ResNet50 (20.5%)	16.9	11.5	8.6	7.4
ResNet101 (24.6%)	25.3	17.3	13.9	13.0
GtObject (100%)	53.9	37.6	33.3	33.3

the diverse variants of activities, complex action dependency patterns, and the variability of scenes. We use it as one of the datasets to validate the ability of our model to handle real-world videos. Charades contains 9,848 crowdsourced videos, with 7,985 training videos and 1,863 testing videos, involves massive interactions with 46 object classes, and has 30 verbs leading to 157 action classes [17]. Meanwhile, Action Genome [18] provides 234,253 frame-level scene graph labels which contain 476,229 object bounding boxes and 1,715,568 relationships for the Charades dataset. These scene graph annotations are used for grammar dictionary learning and action anticipation model training.

Baselines. We compare SEAD with the following baseline methods on the Charades dataset: 1) RandomPred: random prediction; 2) LastPred: it always predicts the last observed frame; 3) FC Autoregressive: it predicts future actions using a fully-connected layer in an autoregressive manner with a time resolution is one second; 4) FC Direct: it directly predicts actions at various future times using a fully-connected layer; 5) LSTM: it predicts the future actions autoregressively using an LSTM network; 6) Grammar [7]: the grammar-based action anticipation model [7]; 7) AGG [8]: Adversarial Generative Grammar, the latest state-of-the-art human action anticipation method on the Charades dataset.

Overall Performance. We obey the same train/test split of videos as the Charades dataset. As shown in Table I, our proposed model SEAD outperforms the deep neural network-based model—LSTM, as well as grammar-based model—Grammar, and the hybrid method—AGG by large margins. In particular, our proposed model SEAD achieves nearly 2x mAP over the state-of-the-art AGG. The reason is that these methods all focus on learning sequential patterns at the action level, and thus they are sensitive to the diverse variants of activities and variability of scenes in the Charades dataset and suffer from significant learning burdens due to the complex action dependency patterns hidden in the Charades dataset. We observe that the durations of most of the videos are around 30 seconds, and only a small fraction of them have lengths exceeding 45 seconds. This suggests that the fluctuation in performance is likely caused by the limited number of videos with lengths in that range.

Table II shows the future 3 seconds prediction results of SEAD regarding the SEQ_LEN, the length of sequence fed into the LSTM. The performance of SEAD first grows with SEQ_LEN and then tends to be stable when it reaches around 4. For this reason, we only feed the last four elements in the sequence to LSTM. Notably, the poor performance of SEAD for SEQ_LEN = 1 indicates the importance of scene graph sequence.

Ablation Study. In this part, we attempt to verify the efficacy of the proposed scene prediction module. To this end, we compare SEAD with its variant SG+LSTM by removing the scene prediction module from it. Specifically, suppose the present moment is t_0, SEAD first predicts the scene graphs between t_0 and $t_0 + \Delta t$ and then takes as input both the observed and predicted scene graphs for the action anticipation module, which outputs actions at time $Y_{t_0+\Delta t}$. In contrast, SG+LSTM makes the action anticipation by solely using the scene graph sequence up to t_0 and actions at time $Y_{t_0+\Delta t}$, in other words, it is equivalent to using an LSTM network to predict future actions by only considering the past observed scene graphs. As shown in Fig. 7, the macro precision, macro recall, and macro F1-score of both methods all drop as Δt increases. Initially, the performance gaps of all three metrics between SEAD and SG+LSTM are small within the first three seconds. However, the gaps become large as Δt continues growing. This is because more uncertainty emerges for a

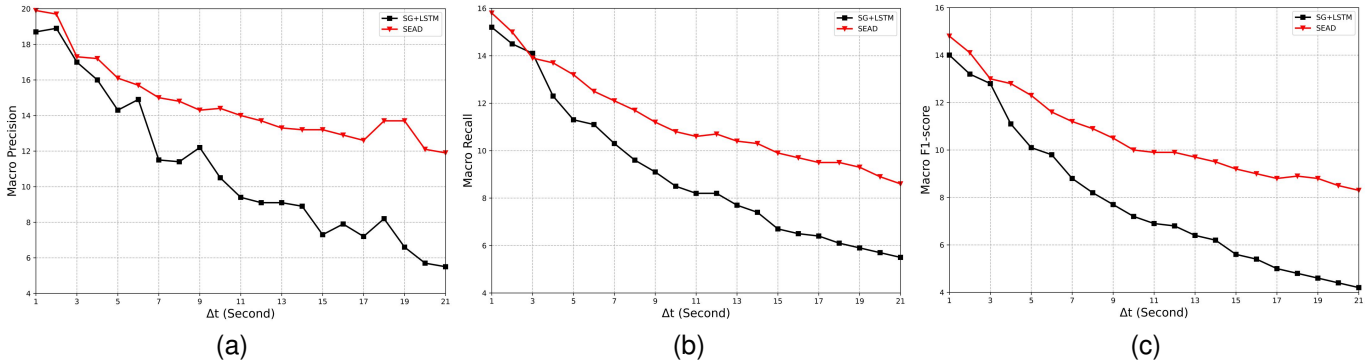


Fig. 7. (a) macro precision of SG+LSTM and SEAD regarding Δt; (b) macro recall of SG+LSTM and SEAD regarding Δt; (c) macro F1-score of SG+LSTM and SEAD regarding Δt.

longer forecasting horizon and it becomes more challenging for the prediction, whereas our proposed scene prediction module is able to predict the most likely scene graphs within Δt that could help to reduce such uncertainty and aid the action anticipation in an effective way.

Additionally, Table III presents the results of our ablation study on Equation 4. The approach labeled as "$\texttt{with\_}o_i$" (equivalent to Equation 4) indicates that the keys in the dictionary $\mathcal{T}$ correspond to (object, event) pairs, with values representing the average duration of events for associated objects. In contrast, "$\texttt{w/o\_}o_i$" refers to the case where the keys of $\mathcal{T}$ exclusively correspond to the spatial-contact events, and the values indicate the average duration of these specific events. The results in Table III demonstrate the superior performance of the "$\texttt{with\_}o_i$" approach, which can be attributed to its more accurate estimation of the duration of spatial-contact events by considering the associated objects.

Moreover, to evaluate the impact of object detection accuracy on our model, we present the future 3-second prediction results utilizing different object detection methods, as summarized in Table IV. These methods include employing two FasterRCNN backbone networks—ResNet50 and ResNet101 (utilized in SEAD)—along with ground truth objects (GtObject). As the table shows, all metrics of SEAD grow with the object detection accuracy (mAP from 20.5% to 100%), which implies a strong positive correlation between the SEAD performance and object detection accuracy.

Qualitative Analysis. Fig. 8 shows a qualitative example of the prediction of SEAD. The model takes as input the video clip from time 0 to t_0 and produces the prediction for the next 4 seconds (at the top of the figure). The predicted scene graphs and actions vs their ground truths at t_0+2 and t_0+4 are shown at the bottom of the figure, the correct prediction and the ground truth are indicated in green color whereas the incorrect prediction is indicated in red one. It can be observed that the predicted scene graph and actions at time t_0+2 are identical to the ground-truth ones, whereas the scene graph is almost correctly predicted at t_0+4, i.e., correct graph structure and most of the nodes, the only discrepancies are two nodes "sitting on" vs "lying on", "not contact" vs "leaning on". Similarly, the predicted actions are also very close to the ground truth ones at t_0+4, the only difference is "Lying on a sofa" vs "Sitting on a sofa", which results from the discrepancies between the predicted scene graph and the ground truth scene graph. This example demonstrates that our proposed method is able to produce meaningful scene graphs and multi-label actions albeit with slight discrepancies. Meanwhile, it also shows that the predicted scene graphs can indeed aid the action prediction, this stands in contrast to the methods focusing on learning the sequential patterns at the action level. In other words, this more intention-aware representation frees SEAD from the burden of recognizing and memorizing the complex dependencies underlying the long action sequences.

C. Action Anticipation on HOMAGE

Dataset. Home Action Genome (HOMAGE) is a relatively new real-world dataset collected by 27 individuals in kitchens, bathrooms, bedrooms, living rooms, and laundry rooms. The dataset is split into 1,388 train videos and 198 test videos with multiple views and modalities. These videos contain 86 object classes and 29 relationship classes and are densely annotated with scene graphs, including 497,534 bounding boxes and 583,481 relationships [20].

Results. To validate the advantages of scene graph level versus the action level for the human action anticipation task, we also conduct an experiment to compare our proposed model SEAD with several aforementioned baselines, as well as with the grammar-based model AOG-Grammar [3] on the real-world HOMAGE dataset. As shown in Table V, our model SEAD achieves an average improvement of 40.6% mAP over the baseline models. Meanwhile, as depicted in Fig. 9, the performance of both SEAD and SG+LSTM on the HOMAGE dataset demonstrate a similar trend on the Charades dataset. It can be observed that SG+LSTM even outperforms the baseline methods. These results imply that learning action anticipation at the scene graph semantic level can indeed boost the prediction performance in comparison to the action level, and also shed light on the great potential of scene graphs in improving the activity inference tasks. The qualitative results on the HOMAGE dataset are very similar to those observed in the Charades dataset, and we do not present them here due to the space limit.

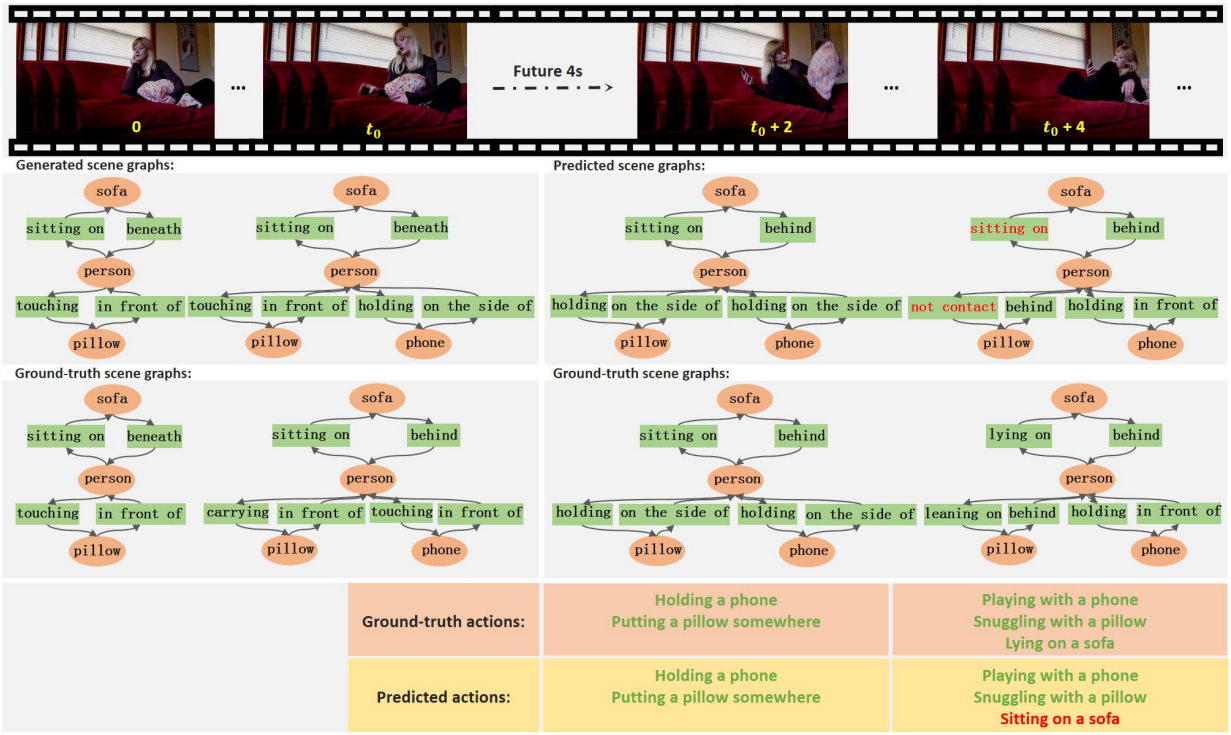


Fig. 8. Qualitative prediction results of future scene graphs and actions using SEAD on the Charades dataset. Ground-truth and correctly predicted actions are shown in green, while red indicates incorrectly predicted actions.

TABLE V
PREDICTION mAP OF ACTIONS WITHIN THE NEXT 10 SECONDS ON THE HOMAGE DATASET

Methods	1s	2s	3s	5s	7s	9s	10s
RandomPred	0.4	0.4	0.4	0.4	0.4	0.4	0.4
AOG-Grammar [3]	7.5	7.1	6.9	6.7	6.5	6.1	5.7
LastPred	13.4	11.7	10.9	9.1	8.3	7.9	7.6
FC Autoregressive	12.3	12.2	11.5	9.4	8.6	7.9	7.5
FC Direct	13.4	12.6	11.9	9.2	8.5	8.1	7.4
LSTM	12.1	12.2	11.6	9.5	8.8	8.2	7.9
SEAD	**16.2**	**15.4**	**15.2**	**14.9**	**13.6**	**12.6**	**11.7**
Improvement (%)	20.9	22.2	27.7	56.8	54.5	53.7	48.1

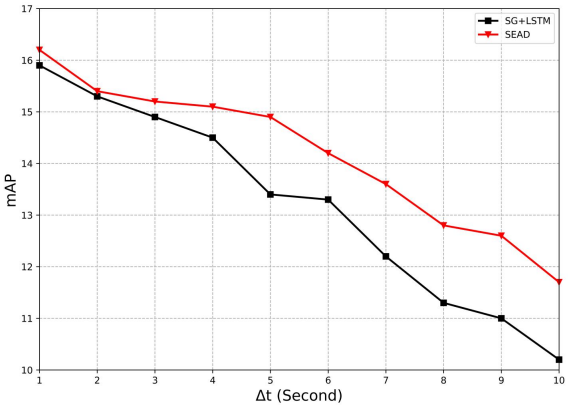


Fig. 9. The mean average precision of SG+LSTM and SEAD regarding Δt.

D. Action Anticipation on CAD-120

Dataset. CAD-120 is a standard dataset used to conduct performance evaluations of many action anticipation models. It contains 120 action sequences of ten different activities performed by four people, where each activity is repeated three times. Each of the activities is a sequence of actions such as moving and opening. Since the CAD-120 dataset has not been annotated with scene graphs, we annotate two typical activities (*arranging_objects* and *having_meal*) with scene graph labels to validate the performance of the SEAD on the standard action anticipation dataset. The scene graph labels are annotated in the same manner as Action Genome [18], which first annotates the objects in the frame with the bounding boxes and then selects the relationship labels from the label set to annotate them.

Results. We predict the actions within the next 3 seconds on two typical activities of the CAD-120 dataset. The results are summarized in Table VI, which shows that all these methods perform relatively well on the activity *arranging_objects* that contains the simplest action sequences in the CAD-120 dataset (the action sequences are almost identical). This implies that the baseline methods indeed have an advantage in dealing with videos that have a clear compositional structure. The accuracy of SEAD drops 6.25% in comparison to its peak performance in *arranging_objects*, this is due to the incorrect predictions of the action "moving box". As illustrated in the top segment of Fig. 10, a node within the scene graph at future time $t_0 + 2$ is mistakenly predicted as "holding" (it should still remain "not contact"). This early prediction of "holding" subsequently leads to the incorrect anticipation of the action "moving box". In other words, the incorrect action anticipation is not caused by the inaccurate prediction of the preceding action, instead, it is due to the incorrect estimation

TABLE VI
PREDICTION RESULTS ON THE CAD-120 DATASET FOR THE FUTURE 3 SECONDS

Methods	*arranging_objects*				*having_meal*			
	Accuracy	Macro Precision	Macro Recall	Macro F1	Accuracy	Macro Precision	Macro Recall	Macro F1
RandomPred	10.0	-	-	-	10.0	-	-	-
LastPred	58.8	62.5	75.0	60.1	36.6	22.2	16.9	18.6
FC Autoregressive	**80.0**	70.0	70.0	57.1	38.2	12.8	20.2	14.6
FC Direct	62.4	55.6	66.2	56.2	44.0	15.0	31.2	17.6
LSTM	**80.0**	66.7	55.6	60.0	38.0	15.9	22.1	18.3
AOG-Grammar [3]	**80.0**	**83.3**	**83.3**	**80.0**	31.3	24.7	21.0	20.6
SG+LSTM	62.5	53.3	55.6	51.7	54.7	39.2	46.2	41.0
SEAD	75.0	53.3	66.7	58.3	**58.5**	**58.9**	**57.0**	**54.8**

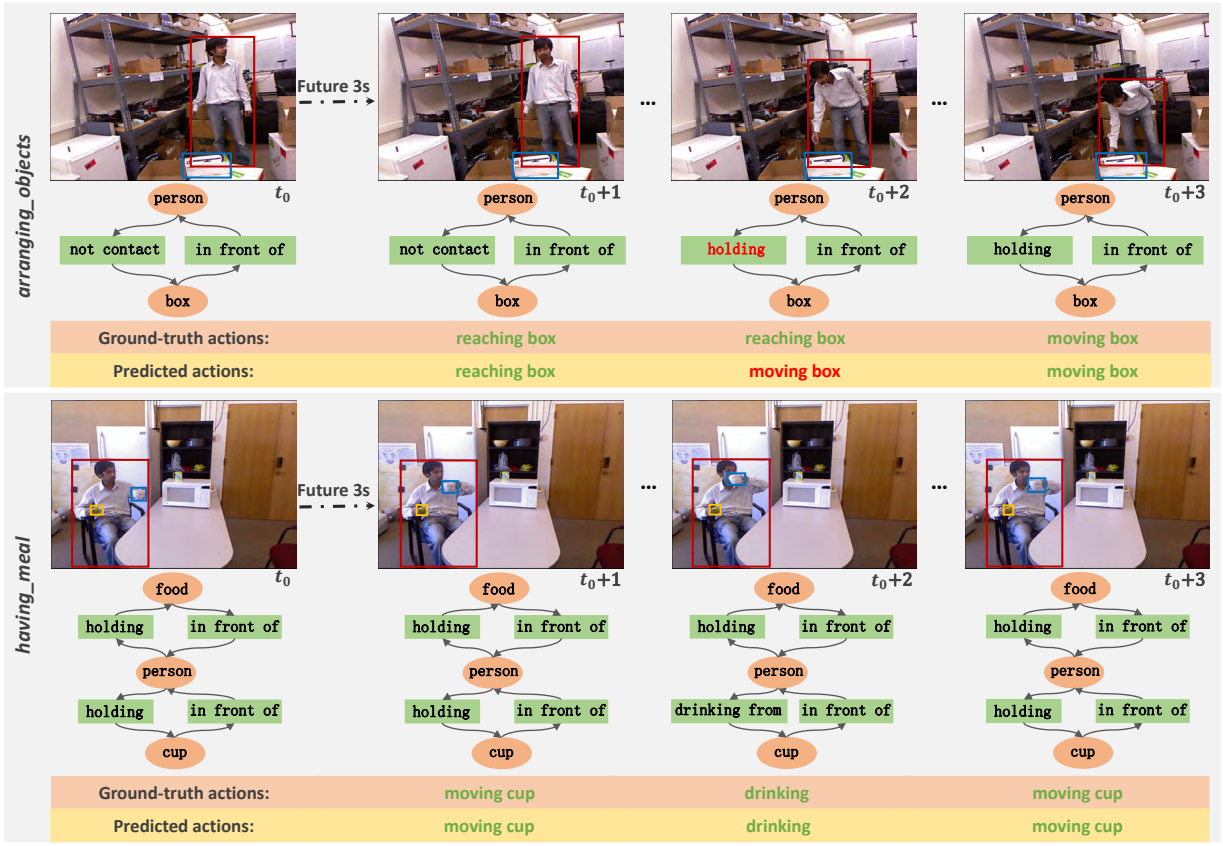


Fig. 10. Qualitative prediction results of future scene graphs and actions using SEAD on the CAD-120 dataset. The ground truth and correctly predicted actions are shown in green, while red indicates incorrectly predicted actions.

of the relationship duration. This indicates that improving the estimation of relationship duration and refining scene graph annotations can further enhance the performance of SEAD.

However, when there are almost no identical action sequences (closer to the real-world scenarios), the performance of the baselines drops significantly on the relatively complex activity *having_meal*; by contrast, SEAD still yields stable performance and outperforms FC Direct by 33% in terms of accuracy. The reason is that these baseline methods all focus on learning sequential patterns at the action level, and thus they are sensitive to the diverse variants of the activity *having_meal*. For instance, after the action "moving cup", a range of potential subsequent actions may occur, including "drinking", "eating", "moving food", "placing cup", and so on. In contrast, the more intention-aware human-object interaction tuple pair frees SEAD from the burden of recognizing the complex dependencies underlying the long action sequences, as shown in the bottom segment of Fig. 10. Moreover, as presented at the bottom of Table VI, in both *arranging_objects* and *having_meal* activities, the superior performance of SEAD over its variant SG+LSTM further verifies the effectiveness of the proposed scene prediction module.

VI. Conclusion

In this paper, we present a scene and action joint prediction model—SEAD—to address the challenges posed by real-world scenarios. In contrast to the existing methods, our proposed model learns the action anticipation at a high semantic level rather than focusing on the low action level. To this end, we propose to utilize the more structural representations—

scene graphs, to capture humans, objects, and their relationships. The rich context semantic information of scene graphs between video frames provides a scaffold for many image processing and computer vision tasks, and to the best of our knowledge, we are the first to bring this powerful representation to action anticipation. The human-object interaction representations are more intention-aware, which frees SEAD from the burden of recognizing and memorizing the complex action dependencies underlying long action sequences and makes SEAD less insensitive to the diverse variants of activities and the variability of scenes in real-world datasets. In SEAD, the scene prediction module predicts future scene graphs using a grammar dictionary that captures the patterns of interactions between humans and each particular object, and the action anticipation module predicts the future actions by using an LSTM network to process the observed and predicted scene graphs.

The experiments demonstrate that our proposed model can achieve desirable prediction results on two real-world datasets and a standard dataset. The excellent performance of learning action anticipation at the high semantic level offers a new opportunity for the prediction models to cope with complex inference tasks. In the future, we plan to improve the performance of action prediction in real-world scenarios regarding the following aspects: 1) The inherent multimodality of action duration in real-world scenarios poses a significant challenge for the task of action anticipation. We aim to address this difficulty by exploring the multi-modal distributions and the dynamic perception approach. 2) Our current work is primarily focused on unimodal RGB images. It is also worthwhile to explore integrating the rich multimodal information in real-world scenarios (such as text and audio) to further enhance model performance. 3) To minimize redundancy in scene graphs across adjacent frames, we will consider generating salient scene graphs for key frames and then integrate it with SEAD to enhance model efficiency.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under Grant No. 62206074 and Grant No. 62072137, Shenzhen College Stability Support Plan under Grant No. GXWD20220811173233001, and the National Key R&D Program of China under Grant No. 2023YFB4503100.

REFERENCES

[1] A. Rasouli, M. Rohani, and J. Luo, "Bifold and semantic reasoning for pedestrian behavior prediction," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 15 600–15 610.

[2] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *International Journal of Computer Vision (IJCV)*, 2022, vol. 130, no. 5, pp. 1366–1401.

[3] S. Qi, S. Huang, P. Wei, and S.-C. Zhu, "Predicting human activities using stochastic grammar," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 1164–1172.

[4] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9925–9934.

[5] S. Qi, B. Jia, S. Huang, P. Wei, and S.-C. Zhu, "A generalized earley parser for human activity parsing and prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020, vol. 43, no. 8, pp. 2538–2554.

[6] Y. B. Ng and B. Fernando, "Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting," *IEEE Transactions on Image Processing (TIP)*, 2020, vol. 29, pp. 8880–8891.

[7] A. Piergiovanni, A. Angelova, and M. S. Ryoo, "Differentiable grammars for videos," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 07, 2020, pp. 11 874–11 881.

[8] A. Piergiovanni, A. Angelova, A. Toshev, and M. S. Ryoo, "Adversarial generative grammars for human activity prediction," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 507–523.

[9] R. Girdhar and K. Grauman, "Anticipative video transformer," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 13 505–13 515.

[10] T.-M. Tai, G. Fiameni, C.-K. Lee, S. See, and O. Lanz, "Unified recurrence modeling for video action anticipation," in *International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 3273–3279.

[11] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2013, pp. 729–738.

[12] H. Kuehne, A. Arslan, and T. Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 780–787.

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

[14] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research (IJRR)*, 2013, vol. 32, no. 8, pp. 951–970.

[15] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1194–1201.

[16] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4362–4370.

[17] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 510–526.

[18] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 236–10 247.

[19] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision (IJCV)*, 2017, vol. 123, no. 1, pp. 32–73.

[20] N. Rai, H. Chen, J. Ji, R. Desai, K. Kozuka, S. Ishizaka, E. Adeli, and J. C. Niebles, "Home action genome: Cooperative compositional action understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 184–11 193.

[21] R. Giegerich, "Introduction to stochastic context free grammars," *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*, 2014, pp. 85–106.

[22] J. Earley, "An efficient context-free parsing algorithm," *Communications of the ACM (CACM)*, 1970, vol. 13, no. 2, pp. 94–102.

[23] S. Holtzen, Y. Zhao, T. Gao, J. B. Tenenbaum, and S.-C. Zhu, "Inferring human intent from video by sampling hierarchical plans," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1489–1496.

[24] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu, "Robot learning with a spatial, temporal, and causal and-or graph," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2144–2151.

[25] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what?-anticipating temporal occurrences of activities," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5343–5352.

[26] A. Furnari and G. M. Farinella, "Rolling-unrolling lstms for action anticipation from first-person video," *IEEE Transactions on Pattern*

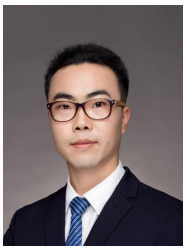
Analysis and Machine Intelligence (TPAMI), 2020, vol. 43, no. 11, pp. 4021–4036.
[27] T. Mahmud, M. Hasan, and A. K. Roy-Chowdhury, "Joint prediction of activity labels and starting times in untrimmed videos," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 5773–5782.
[28] D. Roy and B. Fernando, "Action anticipation using pairwise human-object interactions and transformers," *IEEE Transactions on Image Processing (TIP)*, 2021, vol. 30, pp. 8116–8129.
[29] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5410–5419.
[30] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: an efficient subgraph-based framework for scene graph generation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 335–351.
[31] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 670–685.
[32] V. S. Chen, P. Varma, R. Krishna, M. Bernstein, C. Re, and L. Fei-Fei, "Scene graph prediction with limited labels," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 2580–2590.
[33] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3716–3725.
[34] Y. Cong, W. Liao, H. Ackermann, B. Rosenhahn, and M. Y. Yang, "Spatial-temporal transformer for dynamic scene graph generation," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 16 372–16 382.
[35] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang, "Unpaired image captioning via scene graph alignments," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 10 323–10 332.
[36] K. Nguyen, S. Tripathi, B. Du, T. Guha, and T. Q. Nguyen, "In defense of scene graphs for image captioning," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 1407–1416.
[37] B. Schroeder and S. Tripathi, "Structured query-based image retrieval using scene graphs," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 178–179.
[38] J. Shi, H. Zhang, and J. Li, "Explainable and explicit visual reasoning over scene graphs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8376–8384.
[39] S. V. Nuthalapati, R. Chandradevan, E. Giunchiglia, B. Li, M. Kayser, T. Lukasiewicz, and C. Yang, "Lightweight visual question answering using scene graphs," in *ACM International Conference on Information & Knowledge Management (CIKM)*, 2021, pp. 3353–3357.
[40] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1219–1228.
[41] Y. Li, T. Ma, Y. Bai, N. Duan, S. Wei, and X. Wang, "Pastegan: A semi-parametric method to generate image from scene graph," *Advances in Neural Information Processing Systems (NIPS)*, 2019, vol. 32.
[42] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu, "Modeling 4d human-object interactions for event and object recognition," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 3272–3279.
[43] Z. Solan, D. Horn, E. Ruppin, and S. Edelman, "Unsupervised learning of natural languages," *Proceedings of the National Academy of Sciences (PNAS)*, 2005, vol. 102, no. 33, pp. 11 629–11 634.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

Yuqi Zhang received the M.S. degree from the School of Computer Science and Technology, Northeastern University in 2019. She is currently pursuing her Ph.D. degree at the School of Computer Science and Technology, Harbin Institute of Technology. Her current research interests include video understanding and human action anticipation.

Xiucheng Li received the Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University in 2020. He is an Assistant Professor at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. Before joining HITSZ, he was a research fellow at the School of Computer Science and Engineering, NTU. His current research interests are implicit generative models and time series analysis.

Hao Xie received the Bachelor's degree in software engineering from the Beijing Jiaotong University in 2020. He is currently pursuing the M.S. degree at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interests include scene graph generation and video understanding.

Weijun Zhuang received the M.S. degree from the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. He is currently pursuing the Ph.D. degree at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interests include multimodal learning and vision foundation model.



Shihui Guo is currently an Associate Professor at Xiamen University. He received his Bachelor's degree from Yuanpei College of Peking University in 2010 and Ph.D. from National Computing Animation Center, Bournemouth University in 2015. In 2019, he received the fellowship of Leaders in Innovation from Chinese Academy of Engineering and Royal Academy of Engineering (UK). He has worked as a postdoctoral researcher in the research group of Professor Nadia Thalmann, a member of Swiss Academy of Engineering. His research interests focus on human-computer interaction and computer graphics.

Zhijun Li received the M.S. degree in computer science and technology and the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, in 2001 and 2006, respectively. He is currently a Professor at the School of Computer Science and Technology, Harbin Institute of Technology. His research focuses on wireless networks, Internet of Things, and ubiquitous computing. He was a recipient of the Mobicom17 Best Paper Award.