FUTURE PREDICTION WITH ADVERSARIAL GRAMMARS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose a differentiable adversarial grammar model for future prediction. The objective is to model a formal grammar in terms of differentiable functions and latent representations, so that their learning is possible through standard backpropagation. Learning a formal grammar represented with latent terminals, non-terminals, and productions rules allows capturing sequential structures with multiple possibilities from data.

The adversarial grammar is designed so that it can learn stochastic production rules from the data distribution. Being able to select *multiple* production rules leads to different predicted outcomes, thus efficiently modeling many plausible futures. We confirm the benefit of the adversarial grammar on two diverse tasks: future 3D human pose prediction and future activity prediction. For all settings, the proposed adversarial grammar outperforms the state-of-the-art approaches, being able to predict much more accurately and further in the future, than prior work. Code will be open sourced.

1 INTRODUCTION

Future prediction in videos is one of the most challenging visual tasks. Being able to accurately predict future activities, human or object pose has many important implications, most notably for robot action planning. Prediction is particularly hard because it is not a deterministic process as multiple potential 'futures' are possible, and in the case of human pose, predicting real-valued output vectors is further challenging. Given these challenges, we address the long standing questions: how should the sequential dependencies in the data be modeled and how can multiple possible long-term future outcomes be predicted at any given time.

To address these challenges, we propose an *adversarial grammar* model for future prediction. The model is a differentiable form of a regular grammar trained with adversarial sampling of various possible futures, which is able to output real-valued predictions (e.g., 3D human pose) or semantic prediction (e.g., activity classes). Learning sequences of actions or other sequential processes with the imposed rules of a grammar is valuable, as it imposes temporal structural dependencies and captures relationships between states (e.g., activities). At the same time, the use of adversarial sampling when learning the grammar rules is essential, as this adversarial process is able to produce multiple candidate future sequences that follow a similar distribution to sequences seen in the data. More importantly, a traditional grammar will need to enumerate all possible rules (exponential growth in time) to learn multiple futures. This adversarial stochastic sampling process allows for much more memory-efficient learning without enumeration. Additionally, unlike other techniques for future generation (e.g., autoregressive RNNs), we show the adversarial grammar is able to learn long sequences, can handle multi-label settings, and predict much further into the future.

The proposed approach is driven entirely by the structure imposed from learning grammar rules and their relationships to the terminal symbols of the data and by the adversarial losses which help model the data distribution over long sequences. To our knowledge this is the first approach of adversarial grammar learning and the first to be able to successfully produce *multiple* feasible long-term future predictions for high dimensional outputs.

The approach outperforms previous state-of-the-art methods, including RNN/LSTM and memory based methods. We evaluate future prediction on high dimensional data and are able to predict much further in the future than prior work. The proposed approach is also general – it is applied to

diverse future prediction tasks: 3D human pose prediction and multi-class and multi-label activity forecasting, and on three challenging datasets: Charades, MultiTHUMOS, and Human3.6M.

2 RELATED WORK

Grammar models for visual data. The notion of grammars in computational science was introduced by Chomsky (1956) for description of language, and has found a widespread use in natural language understanding. In the domain of visual data, grammars are used to parse images of scenes (Zhu & Mumford, 2007; Zhao & Zhu, 2011; Han & Zhu, 2008). In their position paper, Zhu & Mumford (2007) present a comprehensive grammar-based language to describe images, and propose MCMC-based inference. More recently, a recursive neural net based approach was applied to parse scenes by Socher et al. (2011). However, this work has no explicit representation of grammar. In the context of temporal visual data, grammars have been applied to activity recognition and parsing (Moore & Essa, 2002; Ryoo & Aggarwal, 2006; Vo & Bobick, 2014; Pirsiavash & Ramanan, 2014) but not to prediction or generation. Qi et al. (2017) used used traditional stochastic grammar to predict activities, but only within 3 seconds.

Generative models for sequences. Generative Adversarial Networks (GANs) are a very powerful mechanism for data generation by an underlying learning of the data distribution through adversarial sampling (Goodfellow et al., 2014). GANs have been very popular for image generation tasks (Emily L Denton, 2015; Isola et al., 2017; Wang et al., 2018; Brock et al., 2019). Prior work on using GANs for improved sequences generation (Yu et al., 2017; Fedus et al., 2018; Hu et al., 2017) has also been successful. Fraccaro et al. (2016) proposed a stochastic RNN which enables generation of different sequences from a given state.

Differentiable Rule Learning Previous approaches that address differentiable rule or grammar learning are most aligned to our work (Yang et al., 2017). However, they can only handle rules with very small branching factors and have not been demonstrated in high dimensional output spaces.

Future pose prediction. Previous approaches for human pose prediction (Fragkiadaki et al., 2015; Ionescu et al., 2014; Tang et al., 2018) are relatively scarce. The dominant theme is the use of recurrent models (RNNs or GRUs/LSTMs) (Fragkiadaki et al., 2015; Martinez et al., 2017). Tang et al. (2018) use attention models specifically to target long-term predictions, up to 1 second in the future. Jain et al. (2016) propose a structural RNN which learns the spatio-temporal relationship of pose joints. The above models, contrary to ours, cannot deal with multi-modality and ambiguity in the predictions, and do not produce multiple futures. These results are also only within short-term horizons and the produced sequences often 'interpolate' actual data examples.

Video Prediction. Without providing an exhaustive survey on video prediction, we note that our approach is related to the video prediction literature (Finn et al., 2016; Denton & Fergus, 2018; Babaeizadeh et al., 2017) where adversarial formulations are also common (Lee et al., 2018).

3 APPROACH

Overview and main insights. Our approach is driven by learning the production rules of a grammar, with which we can learn the transitions between continuous events in time, for example 3D human pose or activity. While an activity or action may be continuous, it can also spawn into many possible futures at different points, similarly to switching between rules in a grammar. For example, an activity corresponding to 'walking' can turn into 'running' or continuing the 'walking' behaviour or change to 'stopping'. These production rules are learned in a differentiable fashion with an adversarial mechanism which allows learning multiple candidate future sequences. This enables robust future prediction, which, more importantly, can easily generate multiple realistic futures.

3.1 DIFFERENTIABLE RULE GENERATION FOR SEQUENCE MODELING

A formal regular grammar is represented as the tuple (N, Σ, P, N_0) where N is a finite non-empty set of non-terminals, Σ is a finite set of terminals (or output symbols), P is a set of production rules, and N_0 is the starting non-terminal symbol, $N_0 \in N$. Productions rules in a regular grammar are of the form $A \rightarrow aB$, $A \rightarrow b$, and $A \rightarrow \epsilon$, where $A, B \in N$, $a, b \in \Sigma$, and ϵ is the empty

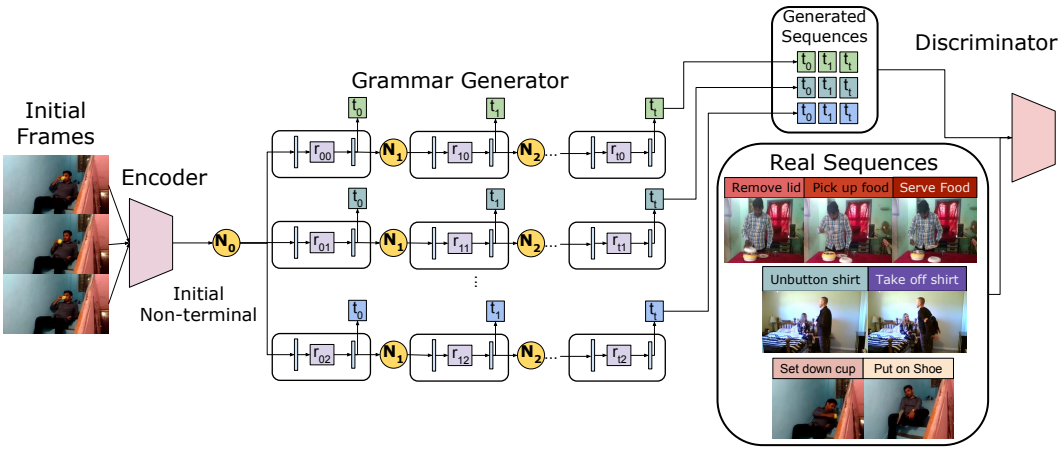

Figure 1: Overview of the adversarial grammar model. The initial non-terminal is produced by an encoder based on some observations. The grammar then generates multiple possible sequences from the non-terminal. The generated and real sequences are used to train the discriminator.

string. Applying multiple productions rules to the starting non-terminal generates a sequence of terminals. Note that we only implement rules of form $A \rightarrow aB$ in our grammar, allowing it to generate sequences infinitely.

Our objective is to learn such non-terminals (e.g., A) and terminals (e.g., a) as latent representations directly from training data, and model the production rules P as a (differentiable) generative neural network function. That is, at the heart of the proposed method is learning nonlinear function $G : N \rightarrow \{(N, \Sigma)\}$ that maps a non-terminal to a *set* of (non-terminal, terminal) pairs. We denote each element (i.e., each production rule) derived from the input non-terminal as $\{(A_i, t_i)\}$. Note that this mapping to multiple possible elements enables modeling of multiple, different sequences, and is not done by existing models (e.g., RNNs).

For any latent non-terminal $A \in N$, the grammar production rules are generated by applying the function G, to A as (here G is a neural network with learnable parameters):

$$\{(B_i, t_i)\}_{i=1:K} = G(A). \tag{1}$$

Each pair corresponds to a particular production rule for this non-terminal. More specifically,

$$\begin{aligned} &A \rightarrow t_1 B_1 \\ &A \rightarrow t_2 B_2 \ldots \\ &A \rightarrow t_K B_K, \text{ where } B_1, B_2, \ldots B_K \in N \end{aligned} \tag{2}$$

This function is applied recursively to obtain a number of output sequences, similar to prior recurrent methods (e.g., RNNs and LSTMs). However, in RNNs, the learned state/memory is required to abstract multiple potential possibilities into a single representation, as the mapping from the state/memory representation to the next representation is deterministic. As a result, when learning from sequential data with multiple possibilities, standard RNNs tend to learn states as a mixture of multiple sequences instead of learning more discriminative states. By learning explicit production rules, our states lead to more salient and distinct predictions which can be exploited for learning long-term, complex output tasks with multiple possibilities, as shown later in the paper.

For example, suppose A is the non-terminal that encodes the activity for 'walking'. An output of the rule $A \rightarrow walkingA$ will be able to generate a sequence of continual 'walking' behavior. Additional rules, e.g., $A \rightarrow stoppingV$, $A \rightarrow runningU$, can be learned, allowing for the activity to switch to 'stopping' or 'running' (with the non-terminals V, U respectively learning to generate their corresponding potential futures). Clearly, for high dimensional outputs, such as 3D human pose, the number and dimensionality of the non-terminals required will be larger. We also note that the non-terminals act as a form of memory, capturing the current state with the Markov property.

To accomplish the above task, G has a special structure. The model contains a number of non-terminals and terminals which are learned: $|N|$ non-terminals of dimensionality D, and $|\Sigma|$ terminals

of dimensionality C (the latter naturally correspond to the number and dimensionality of the desired outputs). G takes input of $A \in N$, then using several nonlinear transformations (e.g., fully connected layers), maps A to a vector r corresponding to a set of rules: $r = f_R(A)$. Here, r is a vector with the size $|P|$ whose elements specify the probability of each rule given input non-terminal. We learn $|P|$ rules which are shared globally, but only a (learned) subset are selected for each non-terminal as the other rule probabilities would become zero. This is conceptually similar to using memory with recurrent neural network methods (Yogatama et al., 2018), but the main difference is that the rule vectors are used to build grammar-like rule structures which are more advantageous in explicitly modeling of temporal dependencies.

In order to generate multiple outputs, the candidate rules, r are followed by the Gumbel-Softmax function (Jang et al., 2017; Maddison et al., 2017), which allows for stochastic selection of a rule. This function is differentiable and samples a single rule from the candidate rules based on the learned rule probabilities. These probabilities model the likelihood of each generated sequence.

Two nonlinear functions f_T and f_N are additionally learned, such that, given a rule r, output the resulting terminal and non-terminal: $B = f_N(r)$, $t = f_T(r)$. These functions are both a sequence of fully-connected layers followed by a non-linear activation function (e.g., softmax or sigmoid depending on the task). As a result, $G(A) = \{(f_N(f_R(A)), f_T(f_R(A)))\}$. The schematic of G is visualized in Figure 1, more details on the functions are provided in the later sections.

The non-terminals and terminals are modeled as sets of high dimensional vectors with pre-specified size and are learned jointly with the rules (all are tunable parameters and naturally more complex datasets require larger capacity). For example, for a simple C-class classification problem, the terminals are represented as C-dimensional vectors matching the one-hot encoding for each class.

Difference to stochastic RNNs Standard recurrent models have a deterministic state, given some input, while the grammar is able to generate multiple potential next non-terminals (i.e., states). Stochastic RNNs (Fraccaro et al., 2016) address this by allowing the next state to be stochastically generated, but this is difficult to control, as the next state now depends on a random value. In the grammar model, the next non-terminal is sampled randomly, but from a set of deterministic candidates. By maintaining a set of deterministic candidates, the next state can be selected randomly or by some other method, giving more control over the generated sequences.

Learning the starting non-terminal. Given an initial input data sequence (e.g., a short video or pose sequences), we learn to generate its corresponding starting non-terminal (i.e., root node). This is used as input to G to generate a sequence of terminal symbols starting from the given non-terminal. Concretely, given the initial input sequence X, a function s is learned which gives the predicted starting non-terminal $N_0 = s(X)$. Then the function G is applied recursively to obtain the possible sequences where j is an index in the sequence and i is one of the possible rules:

$$\begin{cases} \{(B_i^1, t_i^1)\}_i = G(N_0), & j = 0 \\ \{(B_i^{j+1}, t_i^{j+1})\}_i = G(B^j), & \text{for } j > 0 \end{cases} \tag{3}$$

3.2 ADVERSARIAL RULE SAMPLING

The function G generates a set of (non-terminal, terminal) pairs, which is applied recursively to the non-terminals, resulting in new rules and the next set of (non-terminal, terminal) pairs. Note that in most cases, each rule generates a different non-terminal, thus sampling G many times will lead to a variety of generated sequences. As a result, an exponential number of sequences will need to be generated during training, to cover the possible sequences. For example, consider a branching factor of k rules per non-terminal with a sequence of length L. This results in k^L terminals and non-terminals (e.g., for $k = 2$ we have ~ 1000 and for $k = 3 \sim 60,000$). Thus, enumerating all possible sequences is computationally prohibitive beyond $k = 2$. Furthermore, this restricts the tasks that can be addressed to ones with lower dimensional outputs because of memory limits. With $k = 1$ (i.e., no branching), this reduces to a standard RNN during training, unable to generate multiple possible future sequences (i.e., we observed that the rules for each non-terminals become the same).

We address this problem by using stochastic adversarial rule sampling. Given the non-terminals, which effectively contain a number of potential 'futures', we learn *an adversarial-based sampling*,

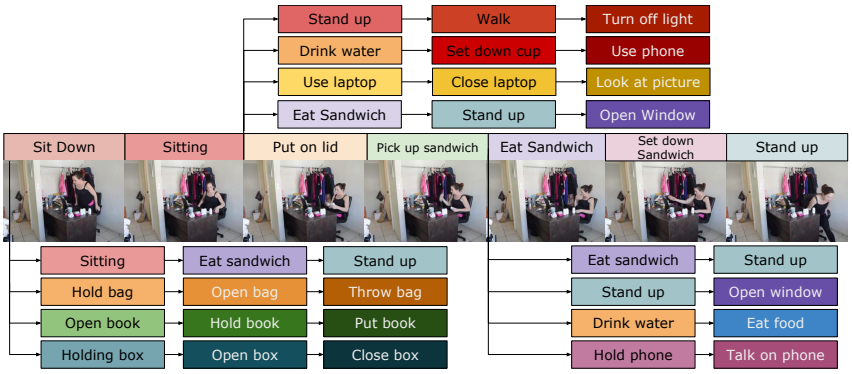

Figure 2: Example video and activity sequence from Charades. At various times, we show multiple futures predicted by the grammar, some matching the true sequence and others very different.

similar to GAN approaches (Goodfellow et al., 2014), which learns to sample the most likely rules for the given input. The use of a discriminator network allows the model to generate realistic sequences that may not match the ground truth without being penalized.

We use the function G, which is the function modeling the learned grammar described above, as the *generator function* and build an additional *discriminator function* D. Following standard GAN training, the discriminator function returns a binary prediction which discriminates examples from the data distribution vs. generated ones. Note that the adversarial process is designed to ultimately generate terminals, i.e., the final output sequence for the model. D is defined as:

$$p = D(n, t) \tag{4}$$

More specifically, D is tasked with the prediction of $p \in \{True, False\}$ based on if the input sequence of terminals, $t = t_0 t_1 t_2 \dots t_L$, is from the data or not (L is the length of the sequence). Note that our discriminator is also conditioned on the non-terminal sequence ($n = n_0 n_1 n_2 \dots n_L$), thus the distribution on non-terminals is learned implicitly, as well.

The discriminator function D is implemented as follows: given an input non-terminal and terminal sequence, we apply several 1D convolutional layers to the terminals and non-terminals, then concatenate their representations followed by a fully-connected layer to produce the binary prediction. (Note that we also tried a GRU/LSTM instead of 1D conv, and it did not making a difference).

The discriminator and generator (grammar) functions are trained to work jointly, as is standard in GANs training. This constitutes the *adversarial grammar*. The optimization objective is defined as:

$$\min_G \max_D = \mathbb{E}_{\mathrm{x} \sim p_{data}(\mathrm{x})}[\log D(\mathrm{x})] + \mathbb{E}_{\mathrm{z} \sim s(X)}[\log(1 - D(G(\mathrm{z})))] \tag{5}$$

where $p_{data}(\mathrm{x})$ is the real data distribution (i.e., sequences of actions or human pose) and $G(\mathrm{z})$ is the generated sequence from an initial state based on a sequence of frames (X).

The sequences generated by G could be compared to the ground truth to compute a loss during training (e.g., maximum likelihood estimation), however, doing so requires enumerating many possibilities in order learn multiple, distinct possible sequences. Without enumeration, the model converges to a mixture representing all possible sequences. By using the adversarial training of G, the model is able to generate sequences that match the distribution observed in the dataset. This allows for computationally feasible learning of longer, higher-dimensional sequences.

Architectures and implementation details. The functions G, f_N and f_t, f_R, mentioned above, are networks using several fully-connected layers, which depend on the task and dataset (specific details are provided in the supplemental material). For pose, the function s is implemented as a two-layer GRU module (Cho et al., 2014) followed by a 1x1 convolutional layer with D_N outputs to produce the starting non-terminal. For activity prediction, s is implemented as two sequential temporal convolutional layers which produce the starting non-terminal. The model is trained for 5000 iterations using gradient descent with momentum of 0.9 and the initial learning rate set to 0.1. We follow the cosine learning rate decay schedule. Our models were trained on a single V100 GPU.

Table 1: Prediction mAP for future activities (higher is better) from 1 seconds to 45 seconds in the future on the MultiTHUMOS dataset.

Method	1 sec	2 sec	5 sec	10 sec	20 sec	30 sec	45 sec
Random	2.6	2.6	2.6	2.6	2.6	2.6	2.6
Last Predicted Action	16.5	16.0	15.1	12.7	8.7	5.8	5.9
FC Autoregressive	17.9	17	14.5	7.7	4.5	4.2	4.7
FC Direct	13.7	9.8	11.0	7.3	8.0	5.5	8.2
LSTM (Autoregressive)	16.5	15.7	12.5	6.8	4.1	3.2	3.9
Grammar only (Ours)	18.7	18.6	13.5	12.8	10.5	8.2	8.5
Adversarial Grammar (Ours)	19.3	19.6	13.1	13.6	11.7	10.4	**11.4**
Adversarial Grammar - max (Ours)	**22.0**	**19.9**	**15.5**	**14.4**	**13.3**	**10.8**	**11.4**

Table 2: Prediction accuracy for future activities for 45 seconds in the future on the Charades dataset.

Method	1 sec	2 sec	5 sec	10 sec	20 sec	30 sec	45 sec
Random	2.4	2.4	2.4	2.4	2.4	2.4	2.4
Last Predicted Action	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 (Autoregressive)	12.6	12.7	12.4	10.8	7.0	6.1	5.4
Grammar only (Ours)	15.7	14.8	12.9	**11.2**	**8.5**	6.6	8.5
Adversarial Grammar (Ours)	15.9	15.0	13.1	10.5	7.4	6.2	8.8
Adversarial Grammar - max (Ours)	**17.0**	**15.9**	**13.4**	10.7	7.8	**7.2**	**9.8**

4 EXPERIMENTS

We conduct experiments on two sets of problems for future prediction: future activity prediction and future 3D human pose prediction and three datasets. Our experiments demonstrate strong performance of the proposed approach over the state-of-the-art and the ability to produce multiple future outcomes, to handle multi-label datasets, and to predict further in the future than prior work.

4.1 ACTIVITY FORECASTING IN VIDEOS

We first test the method for video activity anticipation, where the goal is to predict future activities at various time-horizons, using an initial video sequence as input. We predict future activities up to 45 seconds in the future on well-established video understanding datasets MultiTHUMOS (Yeung et al., 2015) for multi-class prediction and Charades (Sigurdsson et al., 2016) which is a multi-class and multi-label prediction task. We note that we predict much further into the future than prior approaches, that reported results within a second or several seconds (Yeung et al., 2015).

Evaluation metric To evaluate the approaches, we use a standard evaluation metric: we predict the activities occurring T seconds in the future and compute the mean average precision (mAP) between the predictions and ground truth. As the grammar model is able to generate multiple, different future sequences, we also report the maximum mAP over 10 different future predictions. We compare the

Table 3: Evaluation of future pose for short-term and long-term prediction horizons. Measured with Mean Angle Error (lower is better) on Human3.6M. No predictions beyond 1 second are available for prior work. Results are from (Fragkiadaki et al., 2015; Martinez et al., 2017; Tang et al., 2018)

Method	80ms	160ms	320ms	560ms	640ms	720ms	1s	2s	3s	4s
ERD [1]	0.93	1.07	1.31	1.58	1.64	1.70	1.95	-	-	-
LSTM-3LR [1]	0.87	0.93	1.19	1.49	1.55	1.62	1.89	-	-	-
Res-GRU [2]	0.40	0.72	1.09	1.45	1.52	1.59	1.89	-	-	-
Zero-vel. [2]	0.40	0.71	1.07	1.42	1.50	1.57	1.85	-	-	-
MHU-MSE [3]	0.39	0.69	1.04	1.40	1.49	1.57	1.89	-	-	-
MHU [3]	0.39	0.68	1.01	1.34	1.42	**1.49**	1.80	-	-	-
Adv. Gram. (Ours)	**0.36**	**0.65**	**0.98**	**1.27**	**1.40**	**1.49**	**1.74**	**2.25**	**2.70**	**2.98**

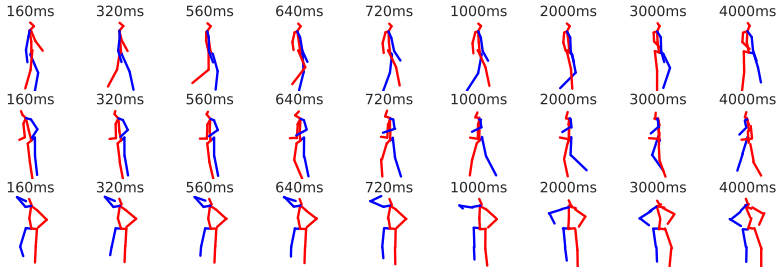

Figure 3: Example results for 3D pose predictions. Top: walking, middle: greeting, bottom: posing.

predictions at 1, 2, 5, 10, 20, 30 and 45 seconds into the future. As little work has explored long-term future activity prediction, we compare against four different baseline methods: (i) repeating the activity prediction of the last seen frame, (ii) using a fully connected layer to predict the next second (applied autoregressively), (iii) using a fully-connected layer to directly predict activities at various future times, and (iv) an LSTM applied autoregressively to future activity predictions.

MultiTHUMOS dataset. The MultiTHUMOS dataset (Yeung et al., 2015) is a popular video understanding benchmark with 400 videos spanning about 30 hours of video and 65 action classes. Table 1 shows activity prediction accuracy for the MultiTHUMOS dataset. In the table, we denote our approach as 'Adversarial Grammar - max' but also report our approach when limited to generating a single outcome ('Adversarial Grammar'), to be consistent to previous methods which are not able to generate more than one outcome. We also compare to grammar without adversarial learning. As seen, our approach outperforms alternative methods including LSTMs. We observe that the gap to other approaches widens further in the future: 3.9 mean accuracy for the LSTM vs 11.2 of ours at 45 seconds in the future, as these autoregressive approaches become noisy. Due to the structure of the grammar model, we are able to generate better long-term predictions. We also find that by predicting multiple futures and taking the max improves performance, confirming that the grammar model is generating different sequences, some of which more closely match the ground truth.

Charades dataset. Charades (Sigurdsson et al., 2016) is a challenging video dataset containing longer-duration activities recorded in home environments. Charades is a multi-label dataset in which multiple activities often co-occur. We use it to demonstrate the ability to handle this complex data. It consists of 9858 videos (7990 training, 1868 test) over 157 activity classes. Table 2 shows the future activity prediction results for Charades. Similarly, we observe that the adversarial grammar model provides more accurate future prediction than previous work, slightly outperformed by grammar only. We note that Charades is more challenging than others on both recognition and prediction tasks, and that grammar only, while performing well here, is not feasible for high dimensional tasks. Figure 2 shows a true sequence and several other sequences generated by the adversarial grammar. As Charades contains many different possible sequences, generating multiple futures is beneficial.

4.2 Human Pose Forecasting

We further evaluate the approach on forecasting 3D human pose, a high dimensional structured-output problem. This is a challenging task (Jain et al., 2016; Fragkiadaki et al., 2015) but is of high importance, e.g., for motion planning in robotics. It also showcases the use of the Adversarial Grammar, as using the standard grammar is not feasible.

Human 3.6M dataset. We conduct experiments on a well established future pose prediction benchmark, the Human 3.6M dataset (Ionescu et al., 2014; Catalin Ionescu, 2011), which has 3.6 million 3D human poses of 15 activities. The goal is to predict the future 3D locations of 32 joints in the human body. We use quaternions to represent each joint location, allowing for a more continuous joint representation space. We also predict differences, rather than absolute positions, which we found leads to more stable learning. Previous work demonstrated prediction results up to a second on this dataset. This work can generate future sequences for longer horizons, 4 seconds in the future.

We compare against the state-of-the-art methods on the Human 3.6M benchmark (Fragkiadaki et al., 2015; Jain et al., 2016; Ionescu et al., 2014; Martinez et al., 2017; Tang et al., 2018) using the Mean Angle Error (MAE) metric as introduced by Jain et al. (2016). Table 3 shows average MAE for all

Table 4: Evaluation of future pose for specific activity classes. Results are Mean Angle Error (lower is better). Results are from (Fragkiadaki et al., 2015; Martinez et al., 2017; Tang et al., 2018). Human3.6M dataset.

Methods	Walking							
	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms
ERD [1]	0.77	0.90	1.12	1.25	1.44	1.45	1.46	1.44
LSTM-3LR [1]	0.73	0.81	1.05	1.18	1.34	1.36	1.37	1.36
Res-GRU [2]	0.27	0.47	0.68	0.76	0.90	0.94	0.99	1.06
Zero-velocity [2]	0.39	0.68	0.99	1.15	1.35	1.37	1.37	1.32
MHU [3]	0.32	0.53	0.69	0.77	0.90	0.94	0.97	1.06
Ours	**0.25**	**0.43**	**0.65**	**0.75**	**0.79**	**0.85**	**0.92**	**0.96**
Methods	Greeting							
	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms
ERD [1]	0.85	1.09	1.45	1.64	1.93	1.89	1.92	1.98
LSTM-3LR [1]	0.80	0.99	1.37	1.54	1.81	1.76	1.79	1.85
Res-GRU [2]	**0.52**	**0.86**	1.30	1.47	1.78	1.75	1.82	1.96
Zero-velocity [2]	0.54	0.89	1.30	1.49	1.79	1.74	1.77	1.80
MHU [3]	0.54	0.87	1.27	**1.45**	1.75	1.71	1.74	1.87
Ours	**0.52**	**0.86**	**1.26**	**1.45**	**1.58**	**1.69**	**1.72**	**1.79**
Methods	Taking photo							
	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms
ERD [1]	0.70	0.78	0.97	1.09	1.20	1.23	1.27	1.37
LSTM-3LR [1]	0.63	0.64	0.86	0.98	1.09	1.13	1.17	1.30
Res-GRU [2]	0.29	0.58	0.90	1.04	1.17	1.23	1.29	1.47
Zero-velocity [2]	0.25	0.51	0.79	0.92	1.03	**1.06**	**1.13**	1.27
MHU [3]	0.27	0.54	0.84	0.96	1.04	1.08	1.14	1.35
Ours	**0.24**	**0.50**	**0.76**	**0.89**	**0.95**	1.08	1.15	**1.24**

Figure 4: Starting from a neutral pose, the grammar is able to generate multiple difference sequences by selecting different rules. Top row: a walking sequence, middle: eating, bottom: sitting.

activities compared to the state-of-the-art methods and Table 4 shows results on several activities, consistent with the protocol in prior work. As seen from the tables, our work outperforms all prior methods. Furthermore, we are able to generate results at larger time horizons of four seconds in the future. In Fig 3, we show some predicted future poses for several different activities, confirming the results reflect the characteristics of the actual behaviors. In Fig. 4, we show the grammar's ability to generate different sequences from a given starting state. Here, given a starting state, we select different rules, which lead to different sequences corresponding to walking, eating or sitting.

5 Conclusion

We propose a novel differentiable adversarial grammar and apply it to several diverse future prediction and generation tasks. Because of the structure we impose for learning grammar-like rules for sequences and learning in adversarial fashion, we are able to generate multiple sequences that follow the distribution seen in data. Our work outperforms prior approaches on all tasks and is able to generate sequences much further in the future. We plan to release the code.

REFERENCES

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *International Conference on Learning Representations (ICLR)*, 2019.

Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

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. *EMNLP*, 2014.

Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

Rob Fergus Emily L Denton, Soumith Chintala. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in Neural Information Processing Systems (NIPS)*, 2015.

W. Fedus, I. Goodfellow, and A. Dai. Maskgan: Better text generation via filling in the _. *International Conference on Learning Representations (ICLR)*, 2018.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 64–72, 2016.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2199–2207, 2016.

K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):59–73, 2008.

Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. Toward controlled generation of text. *International Conference on Machine Learning (ICML)*, 2017.

C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ashesh Jain, Amir Roshan Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

J. Martinez, M. Black, and J. Romero. On human motion prediction using recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Darnell Moore and Irfan Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pp. 770–776, 2002.

Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 612–619, 2014.

Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. Predicting human activities using stochastic grammar. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Michael S Ryoo and Jake K Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1709–1718. IEEE, 2006.

G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136, 2011.

Yongyi Tang, Lin Ma, Wei Liu, and Wei-Shi Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. *IJCAI*, 2018.

Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2641–2648, 2014.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision (IJCV)*, pp. 1–15, 2015.

Dani Yogatama, Yishu Miao, Gabor Melis, Wang Ling, Adhiguna Kuncoro, Chris Dyer, and Phil Blunsom. Memory architectures in recurrent neural network language models. *International Conference on Learning Representations (ICLR)*, 2018.

L. Yu, W. Zhang, J.Wang, and Y. Yu. Seqgan: sequence generative adversarial nets with policy gradient. *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Yibiao Zhao and Song-Chun Zhu. Image parsing with stochastic scene grammar. In *Advances in Neural Information Processing Systems*, pp. 73–81, 2011.

Song-Chun Zhu and David Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2, 2007.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

Activity Prediction For activity prediction, the number of non-terminals ($|N|$) was set to 64, the number of terminals ($|\Sigma|$) was set to the number of classes in the dataset (e.g., 65 in MultiTHUMOS and 157 in Charades). We used 4 rules for each non-terminal (a total of 256 rules). G, f_N and f_t each used one fully connected layer with sizes matching the desired inputs/outputs. s is implemented as a two sequential temporal convolutional layers with 512 channels.

3D Pose estimation For 3D pose, the number of non-terminals ($|N|$) was set to 1024, the number of terminals ($|\Sigma|$) was set to 1024, where each terminal has size of 128 (32 joints in 4D quaternion representation). The number of rules was set to 2 per non-terminal (a total of 2048 rules). G was composed of 2 fully connected layers, f_N and f_t each used three fully connected layers with sizes matching the desired inputs/outputs. s was implemented as a 2-layer GRU using a representation size of 1024.

A.2 SUPPLEMENTAL RESULTS

Table 5 provides results of our approach for future 3D human pose prediction for all activities in the Human3.6M dataset. Figure 5 shows more examples of future predicted 3D pose at different timesteps.

Table 5: Evaluation of future pose of our appoach for both short-term and long-term prediction horizons for all activities. Human3.6M benchmark.

Activity	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1s	2s	3s	4s
Walking	0.25	0.43	0.65	0.75	0.79	0.85	0.92	0.96	1.37	1.34	1.87
Eating	0.2	0.34	0.53	0.67	0.79	0.92	1.01	1.23	1.66	2.01	2.14
Smoking	0.26	0.49	0.92	0.89	0.99	1.01	1.02	1.25	1.95	2.8	3.37
Discussion	0.29	0.65	0.91	1.00	1.23	1.52	1.68	1.93	2.32	2.58	2.65
Directions	0.39	0.59	0.78	0.87	0.99	1.01	1.25	1.46	1.88	2.37	2.19
Greeting	0.52	0.86	1.26	1.45	1.58	1.69	1.72	1.79	2.56	3.08	2.3
Phoning	0.59	1.15	1.51	1.65	1.47	1.71	1.78	1.84	2.63	2.97	3.71
Posing	0.25	0.54	1.19	1.43	1.86	2.10	2.15	2.66	3.46	4.04	4.49
Purchases	0.6	0.85	1.16	1.23	1.58	1.67	1.72	2.4	1.95	2.35	2.63
Sitting	0.39	0.62	1.02	1.17	1.24	1.42	1.48	1.65	2.73	3.09	3.47
SittingDown	0.39	0.75	1.10	1.23	1.35	1.48	1.65	1.88	2.71	3.88	4.81
TakePhoto	0.24	0.5	0.76	0.89	0.95	1.08	1.15	1.24	2.1	2.45	2.72
Waiting	0.31	0.61	1.13	1.37	1.75	1.92	2.12	2.55	2.82	3.18	3.53
WalkingDog	0.54	0.87	1.19	1.35	1.62	1.75	1.82	1.91	2.18	2.83	2.77
WalkTogether	0.25	0.51	0.7	0.74	0.82	0.88	0.91	1.33	1.4	1.62	2.14
Average	0.36	0.65	0.98	1.11	1.27	1.40	1.49	1.74	2.25	2.70	2.98

Figure 5: Various predicted 3D pose sequences for walking, greeting, taking photos, sitting, posing.