

The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction

Junwei Liang^{1*} Lu Jiang² Kevin Murphy² Ting Yu³ Alexander Hauptmann¹
¹Carnegie Mellon University ²Google Research ³Google Cloud AI
 {junweil, alex}@cs.cmu.edu, {lujiang, kpmurphy, yuti}@google.com

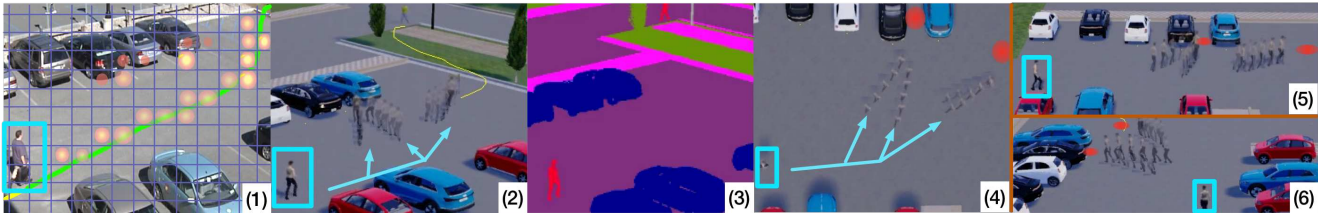


Figure 1: Illustration of person trajectory prediction. (1) A person walks towards a car (data from the VIRAT/ActEV dataset). The green line is the actual future trajectory and the yellow-orange heatmaps are example future predictions. Although these predictions near the cars are plausible, they would be considered errors in the real video dataset. (2) To combat this, we propose a new dataset called “Forking Paths”; here we illustrate 3 possible futures created by human annotators controlling agents in a synthetic world derived from real data. (3) Here we show semantic segmentation of the scene. (4-6) Here we show the same scene rendered from different viewing angles, where the red circles are future destinations.

Abstract

This paper studies the problem of predicting the distribution over multiple possible future paths of people as they move through various visual scenes. We make two main contributions. The first contribution is a new dataset, created in a realistic 3D simulator, which is based on real world trajectory data, and then extrapolated by human annotators to achieve different latent goals. This provides the first benchmark for quantitative evaluation of the models to predict multi-future trajectories. The second contribution is a new model to generate multiple plausible future trajectories, which contains novel designs of using multi-scale location encodings and convolutional RNNs over graphs. We refer to our model as Multiverse. We show that our model achieves the best results on our dataset, as well as on the real-world VIRAT/ActEV dataset (which just contains one possible future).¹

1. Introduction

Forecasting future human behavior is a fundamental problem in video understanding. In particular, future path prediction, which aims at forecasting a pedestrian’s future trajectory in the next few seconds, has received a lot of attention in our community [20, 1, 15, 26]. This functionality is a key component in a variety of applications such as autonomous driving [4, 6], long-term object tracking [19, 48], safety monitoring [30], robotic planning [42, 43], etc.

Of course, the future is often very uncertain: Given the same historical trajectory, a person may take different paths, depending on their (latent) goals. Thus recent work has started focusing on *multi-future trajectory prediction* [53, 6, 26, 34, 54, 23].

Consider the example in Fig. 1. We see a person moving from the bottom left towards the top right of the image, and our task is to predict where he will go next. Since there are many possible future trajectories this person might follow, we are interested in learning a model that can generate multiple plausible futures. However, since the ground truth data only contains one trajectory, it is difficult to evaluate such probabilistic models.

To overcome the aforementioned challenges, our first contribution is the creation of a realistic synthetic dataset that allows us to compare models in a quantitative way in terms of their ability to predict multiple plausible futures, rather than just evaluating them against a single observed trajectory as in existing studies. We create this dataset using the 3D CARLA [11] simulator, where the scenes are manually designed to be similar to those found in the challenging real-world benchmark VIRAT/ActEV [36, 3]. Once we have recreated the static scene, we automatically reconstruct trajectories by projecting real-world data to the 3D simulation world. See Fig. 1 and 3. We then semi-automatically select a set of plausible future destinations (corresponding to semantically meaningful locations in the scene), and ask human annotators to create multiple possible continuations of the real trajectories towards each such goal. In this way, our dataset is “anchored” in reality, and

^{*}Work partially done during a research internship at Google.

¹Code and models are released at <https://next.cs.cmu.edu/multiverse>

yet contains plausible variations in high-level human behavior, which is impossible to simulate automatically.

We call this dataset the “Forking Paths” dataset, a reference to the short story by Jorge Luis Borges.² As shown in Fig. 1, different human annotations have created forkings of future trajectories for the identical historical past. So far, we have collected 750 sequences, with each covering about 15 seconds, from 10 annotators, controlling 127 agents in 7 different scenes. Each agent contains 5.9 future trajectories on average. We render each sequence from 4 different views, and automatically generate dense labels, as illustrated in Fig. 1 and 3. In total, this amounts to 3.2 hours of trajectory sequences, which is comparable to the largest person trajectory benchmark VIRAT/ActEV [3, 36] (4.5 hours), or 5 times bigger than the common ETH/UCY [24, 32] benchmark. We therefore believe this will serve as a benchmark for evaluating models that can predict multiple futures.

Our second contribution is to propose a new probabilistic model, *Multiverse*, which can generate multiple plausible trajectories given the past history of locations and the scene. The model contains two novel design decisions. First, we use a multi-scale representation of locations. In the first scale, the coarse scale, we represent locations on a 2D grid, as shown in Fig. 1(1). This captures high level uncertainty about possible destinations and leads to a better representation of multi-modal distributions. In the second fine scale, we predict a real-valued offset for each grid cell, to get more precise localization. This two-stage approach is partially inspired by object detection methods [41]. The second novelty of our model is to design convolutional RNNs [58] over the spatial graph as a way of encoding inductive bias about the movement patterns of people.

In addition, we empirically validate our model on the challenging real-world benchmark VIRAT/ActEV [36, 3] for single-future trajectory prediction, in which our model achieves the best-published result. On the proposed simulation data for multi-future prediction, experimental results show our model compares favorably against the state-of-the-art models across different settings. To summarize, the main contributions of this paper are as follows: (i) We introduce the first dataset and evaluation methodology that allows us to compare models in a quantitative way in terms of their ability to predict multiple plausible futures. (ii) We propose a new effective model for multi-future trajectory prediction. (iii) We establish a new state of the art result on the challenging VIRAT/ActEV benchmark, and compare various methods on our multi-future prediction dataset.

2. Related Work

Single-future trajectory prediction. Recent works have tried to predict a single best trajectory for pedestrians or vehicles. Early works [35, 59, 62] focused on modeling person

motions by considering them as points in the scene. These research works [21, 60, 33, 30] have attempted to predict person paths by utilizing visual features. Recently Liang *et al.* [30] proposed a joint future activity and trajectory prediction framework that utilized multiple visual features using focal attention [29, 28]. Many works [23, 50, 4, 18, 64] in vehicle trajectory prediction have been proposed. CAR-Net [50] proposed attention networks on top of scene semantic CNN to predict vehicle trajectories. Chauffeur-net [4] utilized imitation learning for trajectory prediction.

Multi-future trajectory prediction. Many works have tried to model the uncertainty of trajectory prediction. Various papers (*e.g.* [20, 43, 44]) use Inverse Reinforcement Learning (IRL) to forecast human trajectories. Social-LSTM [1] is a popular method using social pooling to predict future trajectories. Other works [49, 15, 26, 2] like Social-GAN [15] have utilized generative adversarial networks [14] to generate diverse person trajectories. In vehicle trajectory prediction, DESIRE [23] utilized variational auto-encoders (VAE) to predict future vehicle trajectories. Many recent works [54, 6, 53, 34] also proposed probabilistic frameworks for multi-future vehicle trajectory prediction. Different from these previous works, we present a flexible two-stage framework that combines multi-modal distribution modeling and precise location prediction.

Trajectory Datasets. Many vehicle trajectory datasets [5, 7] have been proposed as a result of self-driving’s surging popularity. With the recent advancement in 3D computer vision research [63, 27, 51, 11, 45, 47, 16], many research works [39, 12, 10, 9, 57, 66, 52] have looked into 3D simulated environment for its flexibility and ability to generate enormous amount of data. We are the first to propose a 3D simulation dataset that is reconstructed from real-world scenarios complemented with a variety of human trajectory continuations for multi-future person trajectory prediction.

3. Methods

In this section, we describe our model for forecasting agent trajectories, which we call *Multiverse*. We focus on predicting the locations of a single agent for multiple steps into the future, $L_{h+1:T}$, given a sequence of past video frames, $V_{1:h}$, and agent locations, $L_{1:h}$, where h is the history length and $T - h$ is the prediction length. Since there is inherent uncertainty in this task, our goal is to design a model that can effectively predict multiple plausible future trajectories, by computing the multimodal distribution $p(L_{h+1:T} | L_{1:h}, V_{1:h})$. See Fig. 2 for a high level summary of the model, and the sections below for more details.

3.1. History Encoder

The encoder computes a representation of the scene from the history of past locations, $L_{1:h}$, and frames, $V_{1:h}$. We encode each ground truth location L_t by an index $Y_t \in G$ representing the nearest cell in a 2D grid G of size $H \times W$,

²https://en.wikipedia.org/wiki/The_Garden_of_Forking_Paths

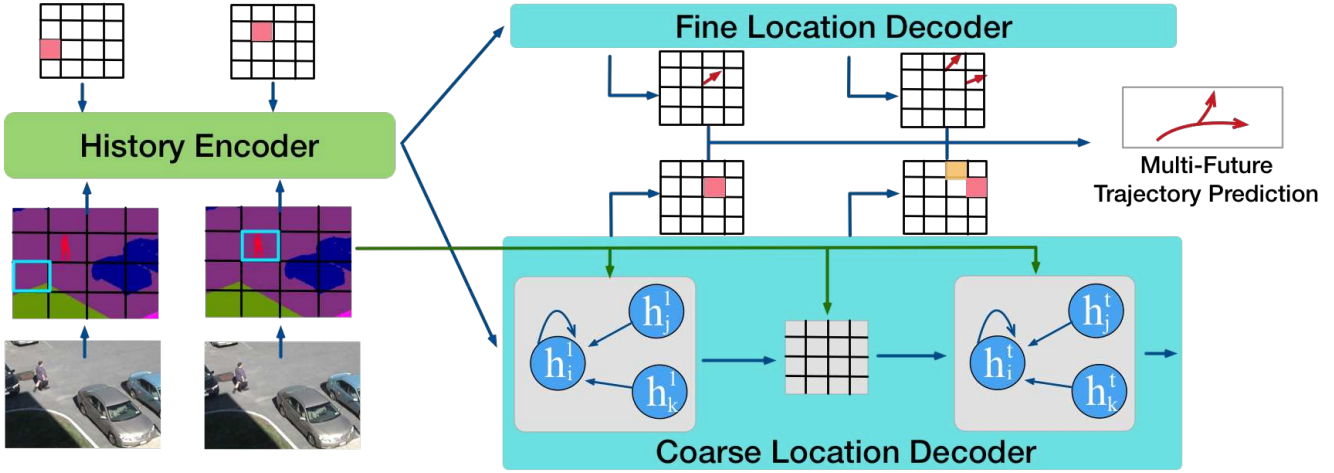


Figure 2: Overview of our model. The input to the model is the ground truth location history, and a set of video frames, which are preprocessed by a semantic segmentation model. This is encoded by the ‘‘History Encoder’’ convolutional RNN. The output of the encoder is fed to the convolutional RNN decoder for location prediction. The coarse location decoder outputs a heatmap over the 2D grid of size $H \times W$. The fine location decoder outputs a vector offset within each grid cell. These are combined to generate a multimodal distribution over \mathbb{R}^2 for predicted locations.

indexed from 1 to HW . Inspired by [22, 31], we encode location with two different grid scales (36×18 and 18×9); we show the benefits of this multi-scale encoding in Section 5.4. For simplicity of presentation, we focus on a single $H \times W$ grid.

To make the model more invariant to low-level visual details, and thus more robust to domain shift (e.g., between different scenes, different views of the same scene, or between real and synthetic images), we preprocess each video frame V_t using a pre-trained semantic segmentation model, with $K = 13$ possible class labels per pixel. We use the Deeplab model [8] trained on the ADE20k [65] dataset, and keep its weights frozen. Let S_t be this semantic segmentation map modeled as a tensor of size $H \times W \times K$.

We then pass these inputs to a convolutional RNN [58, 56] to compute a spatial-temporal feature history:

$$H_t^e = \text{ConvRNN}(\text{one-hot}(Y_t) \odot (W * S_t), H_{t-1}^e) \quad (1)$$

where \odot is element wise product, and $*$ represents 2D-convolution. The function $\text{one-hot}(\cdot)$ projects a cell index into an one-hot embedding of size $H \times W$ according to its spatial location. We use the final state of this encoder $H_t^e \in \mathbb{R}^{H \times W \times d_{enc}}$, where d_{enc} is the hidden size, to initialize the state of the decoders. We also use the temporal average of the semantic maps, $\bar{S} = \frac{1}{h} \sum_{t=1}^h S_t$, during each decoding step. The context is represented as $\mathcal{H} = [H_h^e, \bar{S}]$.

3.2. Coarse Location Decoder

After getting the context \mathcal{H} , our goal is to forecast future locations. We initially focus on predicting locations at the level of grid cells, $Y_t \in G$. In Section 3.3, we discuss how to predict a continuous offset in \mathbb{R}^2 , which specifies a ‘‘delta’’ from the center of each grid cell, to get a fine-grained location prediction.

Let the coarse distribution over grid locations at time t

(known as the ‘‘belief state’’) be denoted by $C_t(i) = p(Y_t = i | Y_{h:t-1}, \mathcal{H})$, for $\forall i \in G$ and $t \in [h+1, T]$. For brevity, we use a single index i to represent a cell in the 2D grid. Rather than assuming a Markov model, we update this using a convolutional recurrent neural network, with hidden states H_t^C . We then compute the belief state by:

$$C_t = \text{softmax}(W * H_t^C) \in \mathbb{R}^{HW} \quad (2)$$

Here we use 2D-convolution with one filter and flatten the spatial dimension before applying softmax. The hidden state is updated using:

$$H_t^C = \text{ConvRNN}(\text{GAT}(H_{t-1}^C), \text{embed}(C_{t-1})) \quad (3)$$

where $\text{embed}(C_{t-1})$ embeds into a 3D tensor of size $H \times W \times d_e$ and d_e is the embedding size. $\text{GAT}(H_{t-1}^C)$ is a graph attention network [55], where the graph structure corresponds to the 2D grid in G . More precisely, let h_i be the feature vector corresponding to the i -th grid cell in H_{t-1}^C , and let \tilde{h}_i be the corresponding output in $\tilde{H}_{t-1}^C = \text{GAT}(H_{t-1}^C) \in \mathbb{R}^{H \times W \times d_{dec}}$, where d_{dec} is the size of the decoder hidden state. We compute these outputs of GAT using:

$$\tilde{h}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} f_e([v_i, v_j]) + h_i \quad (4)$$

where \mathcal{N}_i are the neighbors of node v_i in G with each node represented as $v_i = [h_i, \bar{S}_i]$, where \bar{S}_i collects the cell i ’s feature in \bar{S} . f_e is some edge function (implemented as an MLP in our experiments) that computes the attention weights.

The graph-structured update function for the RNN ensures that the probability mass ‘‘diffuses out’’ to nearby grid cells in a controlled manner, reflecting the prior knowledge that people do not suddenly jump between distant locations. This inductive bias is also encoded in the convolutional

structure, but adding the graph attention network gives improved results, because the weights are input-dependent and not fixed.

3.3. Fine Location Decoder

The 2D heatmap is useful for capturing multimodal distributions, but does not give very precise location predictions. To overcome this, we train a second convolutional RNN decoder H_t^O to compute an offset vector for each possible grid cell using a regression output, $O_t = \text{MLP}(H_t^O) \in \mathbb{R}^{H \times W \times 2}$. This RNN is updated using

$$H_t^O = \text{ConvRNN}(\text{GAT}(H_{t-1}^O), O_{t-1}) \in \mathbb{R}^{H \times W \times d_{dec}} \quad (5)$$

To compute the final prediction location, we first flatten the spatial dimension of O_t into $\tilde{O}_t \in \mathbb{R}^{HW \times 2}$. Then we use

$$L_t = Q_i + \tilde{O}_{ti} \quad (6)$$

where i is the index of the selected grid cell, $Q_i \in \mathbb{R}^2$ is the center of that cell, and $\tilde{O}_{ti} \in \mathbb{R}^2$ is the predicted offset for that cell at time t . For single-future prediction, we use greedy search, namely $i = \text{argmax}_i C_t$ over the belief state. For multi-future prediction, we use beam search in Section 3.5.

This idea of combining classification and regression is partially inspired by object detection methods (e.g., [41]). It is worth noting that in concurrent work, [6] also designed a two-stage model for trajectory forecasting. However, their classification targets are pre-defined anchor trajectories. Ours is not limited by the predefined anchors.

3.4. Training

Our model trains on the observed trajectory from time 1 to h and predicts the future trajectories (in xy -coordinates) from time $h+1$ to T . We supervise this training by providing ground truth targets for both the heatmap (belief state), C_t^* , and regression offset map, O_t^* . In particular, for the coarse decoder, the cross-entropy loss is used:

$$\mathcal{L}_{cls} = -\frac{1}{T} \sum_{t=h+1}^T \sum_{i \in G} C_{ti}^* \log(C_{ti}) \quad (7)$$

For the fine decoder, we use the smoothed L_1 loss used in object detection [41]:

$$\mathcal{L}_{reg} = \frac{1}{T} \sum_{t=h+1}^T \sum_{i \in G} \text{smooth}_{L_1}(O_{ti}^*, O_{ti}) \quad (8)$$

where $O_{ti}^* = L_t^* - Q_i$ is the delta between the true location and the center of the grid cell at i and L_t^* is the ground truth for L_t in Eq.(6). We impose this loss on every cell to improve the robustness.

The final loss is then calculated using

$$\mathcal{L}(\theta) = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{reg} + \lambda_2 \|\theta\|_2^2 \quad (9)$$

where λ_2 controls the ℓ_2 regularization (weight decay), and $\lambda_1 = 0.1$ is used to balance the regression and classification losses.

Note that during training, when updating the RNN, we feed in the predicted soft distribution over locations, C_t . See Eq. (2). An alternative would be to feed in the true values, C_t^* , i.e., use teacher forcing. However, this is known to suffer from problems [40].

3.5. Inference

To generate multiple qualitatively distinct trajectories, we use the diverse beam search strategy from [25]. To define this precisely, let B_{t-1} be the beam at time $t-1$; this set contains K trajectories (history selections) $M_{t-1}^k = \{\hat{Y}_1^k, \dots, \hat{Y}_{t-1}^k\}$, $k \in [1, K]$, where \hat{Y}_t^k is an index in G , along with their accumulated log probabilities, P_{t-1}^k . Let $C_t^k = f(M_{t-1}^k) \in \mathbb{R}^{HW}$ be the coarse location output probability from Eq. (2) and (3) at time t given inputs M_{t-1}^k .

The new beam is computed using

$$B_t = \text{topK}(\{P_{t-1}^k + \log(C_t^k(i)) + \gamma(i) | \forall i \in G, k \in [1, K]\}) \quad (10)$$

where $\gamma(i)$ is a diversity penalty term, and we take the top K elements from the set produced by considering values with $k = 1 : K$. If $K = 1$, this reduces to greedy search.

Once we have computed the top K future predictions, we add the corresponding offset vectors to get K trajectories by $L_t^k \in \mathbb{R}^2$. This constitutes the final output of our model.

4. The Forking Paths Dataset

In this section, we describe our human-annotated simulation dataset, called Forking Paths, for multi-future trajectory evaluation.

Existing datasets. There are several real-world datasets for trajectory evaluation, such as SDD [46], ETH/UCY [37, 24], KITTI [13], nuScenes [5] and VIRAT/ActEV [3, 36]. However, they all share the fundamental problem that one can only observe one out of many possible future trajectories sampled from the underlying distribution. This is broadly acknowledged in prior works [34, 54, 6, 15, 44, 43] but has not yet been addressed.

The closest work to ours is the simulation used in [34, 54, 6]. However, these only contain artificial trajectories, not human generated ones. Also, they use a highly simplified 2D space, with pedestrians oversimplified as points and vehicles as blocks; no other scene semantics are provided.

Reconstructing reality in simulator. In this work, we use CARLA [11], a near-realistic open source simulator built on top of the Unreal Engine 4. Following prior simulation datasets [12, 47], we *semi-automatically* reconstruct static scenes and their dynamic elements from the real-world videos in ETH/UCY and VIRAT/ActEV. There are 4 scenes in ETH/UCY and 5 in VIRAT/ActEV. We exclude 2 cluttered scenes (UNIV & 0002) that we are not able to reconstruct in CARLA, leaving 7 static scenes in our dataset.

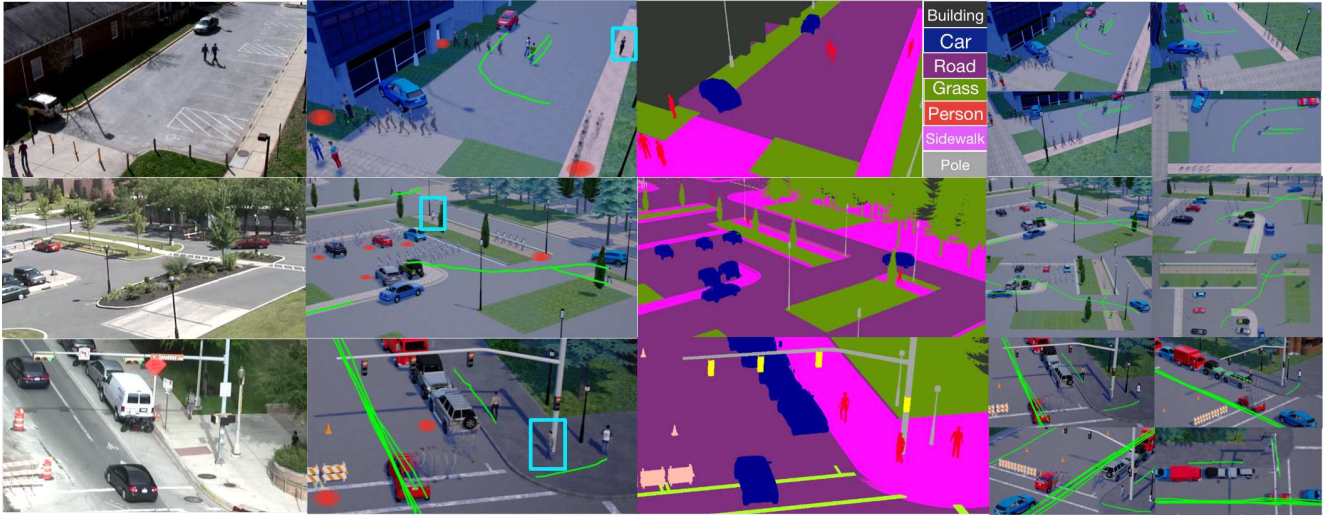


Figure 3: Visualization of the Forking Paths dataset. On the left is examples of the real videos and the second column shows the reconstructed scenes. The person in the blue bounding box is the controlled agent and multiple future trajectories annotated by humans are shown by overlaid person frames. The red circles are the defined destinations. The green trajectories are future trajectories of the reconstructed uncontrolled agents. The scene semantic segmentation ground truth is shown in the third column and the last column shows all four camera views including the top-down view.

For dynamic movement of vehicle and pedestrian, we first convert the ground truth trajectory annotations from the real-world videos to the ground plane using the provided homography matrices. We then match the real-world trajectories’ origin to correct locations in the re-created scenes.

Human generation of plausible futures. We manually select sequences with more than one pedestrian. We also require that at least one pedestrian could have multiple plausible alternative destinations. We insert plausible pedestrians into the scene to increase the diversity of the scenarios. We then select one of the pedestrians to be the “controlled agent” (CA) for each sequence, and set meaningful destinations within reach, like a car or an entrance of a building. On average, each agent has about 3 destinations to move towards. In total, we have 127 CAs from 7 scenes. We call each CA and their corresponding scene a scenario.

For each scenario, there are on average 5.9 human annotators to control the agent to the defined destinations. Specifically, they are asked to watch the first 5 seconds of video, from a first-person view (with the camera slightly behind the pedestrian) and/or an overhead view (to give more context). They are then asked to control the motion of the agent so that it moves towards the specified destination in a “natural” way, *e.g.*, without colliding with other moving objects (whose motion is derived from the real videos, and is therefore unaware of the controlled agent). The annotation is considered successful if the agent reached the destination without colliding within the time limit of 10.4 seconds. All final trajectories in our dataset are examined by humans to ensure reliability.

Note that our videos are up to 15.2 seconds long. This is slightly longer than previous works (*e.g.* [1, 15, 30, 49, 26, 62, 64]) that use 3.2 seconds of observation and 4.8 seconds

for prediction. (We use 10.4 seconds for the future to allow us to evaluate longer term forecasts.)

Generating the data. Once we have collected human-generated trajectories, 750 in total after data cleaning, we render each one in four camera views (three 45-degree and one top-down view). Each camera view has 127 scenarios in total and each scenario has on average 5.9 future trajectories. With CARLA, we can also simulate different weather conditions, although we did not do so in this work. In addition to agent location, we collect ground truth for pixel-precise scene semantic segmentation from 13 classes including sidewalk, road, vehicle, pedestrian, *etc.* See Fig. 3.

5. Experimental results

This section evaluates various methods, including our *Multiverse* model, for multi-future trajectory prediction on the proposed Forking Paths dataset. To allow comparison with previous works, we also evaluate our model on the challenging VIRAT/ActEV [3, 36] benchmark for single-future path prediction.

5.1. Evaluation Metrics

Single-Future Evaluation. In real-world videos, each trajectory only has one sample of the future, so models are evaluated on how well they predict that single trajectory. Following prior work [30, 1, 15, 49, 23, 18, 6, 44], we introduce two standard metrics for this setting.

Let $Y^i = Y_{t=(h+1) \dots T}^i$ be the ground truth trajectory of the i -th sample, and \hat{Y}^i be the corresponding prediction. We then employ two distance-based error metrics: i) *Average Displacement Error* (ADE): the average Euclidean distance between the ground truth coordinates and the prediction co-

ordinates over all time instants:

$$\text{ADE} = \frac{\sum_{i=1}^N \sum_{t=h+1}^T \|Y_t^i - \hat{Y}_t^i\|_2}{N \times (T - h)} \quad (11)$$

ii) *Final Displacement Error* (FDE): the Euclidean distance between the predicted points and the ground truth point at the final prediction time:

$$\text{FDE} = \frac{\sum_{i=1}^N \|Y_T^i - \hat{Y}_T^i\|_2}{N} \quad (12)$$

Multi-Future Evaluation. Let $Y^{ij} = Y_{t=(h+1)\dots T}^{ij}$ be the j -th true future trajectory for the i -th test sample, for $\forall j \in [1, J]$, and let \hat{Y}^{ik} be the k 'th sample from the predicted distribution over trajectories, for $k \in [1, K]$. Since there is no agreed-upon evaluation metric for this setting, we simply extend the above metrics, as follows: i) *Minimum Average Displacement Error Given K Predictions* (minADE_K): similar to the metric described in [6, 43, 44, 15], for each true trajectory j in test sample i , we select the closest overall prediction (from the K model predictions), and then measure its average error:

$$\text{minADE}_K = \frac{\sum_{i=1}^N \sum_{j=1}^J \min_{k=1}^K \sum_{t=h+1}^T \|Y_t^{ij} - \hat{Y}_t^{ik}\|_2}{N \times (T - h) \times J} \quad (13)$$

ii) *Minimum Final Displacement Error Given K Predictions* (minFDE_K): similar to minADE_K, but we only consider the predicted points and the ground truth point at the final prediction time instant:

$$\text{minFDE}_K = \frac{\sum_{i=1}^N \sum_{j=1}^J \min_{k=1}^K \|Y_T^{ij} - \hat{Y}_T^{ik}\|_2}{N \times J} \quad (14)$$

iii) *Negative Log-Likelihood* (NLL): Similar to NLL metrics used in [34, 6], we measure the fit of ground-truth samples to the predicted distribution.

5.2. Multi-Future Prediction on Forking Paths

Dataset & Setups. The proposed Forking Paths dataset in Section 4 is used for multi-future trajectory prediction evaluation. Following the setting in previous works [30, 1, 15, 1, 15, 49, 34], we downsample the videos to 2.5 fps and extract person trajectories using code released in [30], and let the models observe 3.2 seconds (8 frames) of the controlled agent before outputting trajectory coordinates in the pixel space. Since the length of the ground truth future trajectories are different, each model needs to predict the maximum length at test time but we evaluate the predictions using the actual length of each true trajectory.

Baseline methods. We compare our method with two simple baselines, and three recent methods with released source code, including a recent model for multi-future prediction and the state-of-the-art model for single-future prediction: **Linear** is a single layer model that predicts the next coordinates using a linear regressor based on the previous input point. **LSTM** is a simple LSTM [17]

encoder-decoder model with coordinates input only. **Social LSTM** [1]: We use the open source implementation from (<https://github.com/agrimgupta92/sgan/>). **Next** [30] is the state-of-the-art method for single-future trajectory prediction on the VIRAT/ActEV dataset. We train the Next model without the activity labels for fair comparison using the code from (<https://github.com/google/next-prediction/>). **Social GAN** [15] is a recent multi-future trajectory prediction model trained using Minimum over N (MoN) loss. We train two model variants (called PV and V) detailed in the paper using the code from [15].

All models are trained on real videos (from VIRAT/ActEV – see Section 5.3 for details) and tested on our synthetic videos (with CARLA-generated pixels, and annotator-generated trajectories). Most models just use trajectory data as input, except for our model (which uses trajectory and semantic segmentation) and Next (which uses trajectory, bounding box, semantic segmentation, and RGB frames).

Implementation Details. We use ConvLSTM [58] cell for both the encoder and decoder. The embedding size is set to 32, and the hidden sizes for the encoder and decoder are both 256. The scene semantic segmentation features are extracted from the deeplab model [8], pretrained on the ADE-20k [65] dataset. We use Adadelta optimizer [61] with an initial learning rate of 0.3 and weight decay of 0.001. Other hyper-parameters for the baselines are the same to the ones in [15, 30]. We evaluate the top $K = 20$ predictions for multi-future trajectories. For the models that only output a single trajectory, including Linear, LSTM, Social-LSTM, and Next, we duplicate the output for K times before evaluating. For Social-GAN, we use K different random noise inputs to get the predictions. For our model, we use diversity beam search [25, 38] as described in Section 3.5.

Quantitative Results. Table 1 lists the multi-future evaluation results, where we divide the evaluation according to the viewing angle of camera, 45-degree vs. top-down view. We repeat all experiments (except “linear”) 5 times with random initialization to produce the mean and standard deviation values. As we see, our model outperforms baselines in all metrics and it performs significantly better on the minADE metric, which suggests better prediction quality over all time instants. Notably, our model outperforms Social GAN by a large margin of at least 8 points on all metrics. We also measure the standard negative log-likelihood (NLL) metric for the top methods in Table 2.

Qualitative analysis. We visualize some outputs of the top 4 methods in Fig. 4. In each image, the yellow trajectories are the history trajectory of each controlled agent (derived from real video data) and the green trajectories are the ground truth future trajectories from human annotators. The predicted trajectories are shown in yellow-orange heatmaps for multi-future prediction methods, and in red lines for

Method	Input Types	minADE ₂₀		minFDE ₂₀	
		45-degree	top-down	45-degree	top-down
Linear	Traj.	213.2	197.6	403.2	372.9
LSTM	Traj.	201.0 \pm 2.2	183.7 \pm 2.1	381.5 \pm 3.2	355.0 \pm 3.6
Social-LSTM [1]	Traj.	197.5 \pm 2.5	180.4 \pm 1.0	377.0 \pm 3.6	350.3 \pm 2.3
Social-GAN (PV) [15]	Traj.	191.2 \pm 5.4	176.5 \pm 5.2	351.9 \pm 11.4	335.0 \pm 9.4
Social-GAN (V) [15]	Traj.	187.1 \pm 4.7	172.7 \pm 3.9	342.1 \pm 10.2	326.7 \pm 7.7
Next [30]	Traj.+Bbox+RGB+Seg.	186.6 \pm 2.7	166.9 \pm 2.2	360.0 \pm 7.2	326.6 \pm 5.0
Ours	Traj.+Seg.	168.9 \pm 2.1	157.7 \pm 2.5	333.8 \pm 3.7	316.5 \pm 3.4

Table 1: Comparison of different methods on the Forking Paths dataset. Lower numbers are better. The numbers for the column labeled “45 degrees” are averaged over 3 different 45-degree views. For the input types, “Traj.”, “RGB”, “Seg.” and “Bbox.” mean the inputs are xy coordinates, raw frames, semantic segmentations and bounding boxes of all objects in the scene, respectively. All models are trained on real VIRAT/ActEV videos and tested on synthetic (CARLA-rendered) videos.

Method	$T_{pred} = 1$	$T_{pred} = 2$	$T_{pred} = 3$
(PV) [14]	10.08 \pm 0.25	17.28 \pm 0.42	23.34 \pm 0.47
(V) [14]	9.95 \pm 0.35	17.38 \pm 0.49	23.24 \pm 0.54
Next [27]	8.32 \pm 0.10	14.98 \pm 0.19	22.71 \pm 0.11
Ours	2.22 \pm 0.54	4.46 \pm 1.33	8.14 \pm 2.81

Table 2: Negative Log-likelihood comparison of different methods on the Forking Paths dataset. For methods that output multiple trajectories, we quantize the xy -coordinates into the same grid as our method and get a normalized probability distribution prediction.

single-future prediction methods. As we see, our model correctly generally puts probability mass where there is data, and does not “waste” probability mass where there is no data.

Error analysis. We show some typical errors our model makes in Fig. 5. The first image shows our model misses the correct direction, perhaps due to lack of diversity in our sampling procedure. The second image shows our model sometimes predicts the person will “go through” the car (diagonal red beam) instead of going around it. This may be addressed by adding more training examples of “going around” obstacles. The third image shows our model predicts the person will go to a moving car. This is due to the lack of modeling of the dynamics of other far-away agents in the scene. The fourth image shows a hard case where the person just exits the vehicle and there is no indication of where they will go next (so our model “backs off” to a sensible “stay nearby” prediction). We leave solutions to these problems to future work.

5.3. Single-Future Prediction on VIRAT/ActEV

Dataset & Setups. NIST released VIRAT/ActEV [3] for activity detection research in streaming videos in 2018. This dataset is a new version of the VIRAT [36] dataset, with more videos and annotations. The length of videos with publicly available annotations is about 4.5 hours. Following [30], we use the official training set for training and the official validation set for testing. Other setups are the same as in Section 5.2, except we use the single-future eval-

uation metric.

Quantitative Results. Table 3 (first column) shows the evaluation results. As we see, our model achieves state-of-the-art performance. The improvement is especially large on Final Displacement Error (FDE) metric, attributing to the coarse location decoder that helps regulate the model prediction for long-term prediction. The gain shows that our model does well at both single future prediction (on real data) and multiple future prediction on our quasi-synthetic data.

Generalizing from simulation to real-world. As described in Section 4, we generate simulation data first by reconstructing from real-world videos. To verify the quality of the reconstructed data, and the efficacy of learning from simulation videos, we train all the models on the simulation videos derived from the real data. We then evaluate on the real test set of VIRAT/ActEV. As we see from the right column in Table 3, all models do worse in this scenario, due to the difference between synthetic and real data. We find the performance ranking of different methods are consistent between the real and our simulation training data. This suggests the errors mainly coming from the model, and substantiates the rationality of using the proposed dataset to compare the relative performance of different methods.

There are two sources of error. The synthetic trajectory data only contains about 60% of the real trajectory data, due to difficulties reconstructing all the real data in the simulator. In addition, the synthetic images are not photo realistic. Thus methods (such as Next [30]) that rely on RGB input obviously suffer the most, since they have never been trained on “real pixels”. Our method, which uses trajectories plus high level semantic segmentations (which transfers from synthetic to real more easily) suffers the least drop in performance, showing its robustness to “domain shift”. See Table 1 for input source comparison between methods.

5.4. Ablation Experiments

We test various ablations of our model on both the single-future and multi-future trajectory prediction to substantiate our design decisions. Results are shown in Ta-

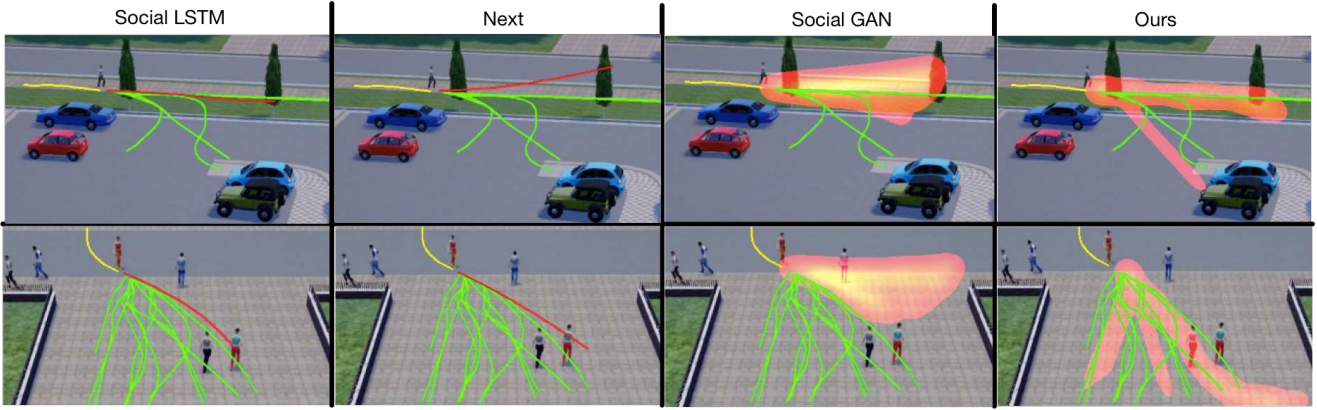


Figure 4: Qualitative analysis. The red trajectories are single-future method predictions and the yellow-orange heatmaps are multi-future method predictions. The yellow trajectories are observations and the green ones are ground truth multi-future trajectories. See text for details.

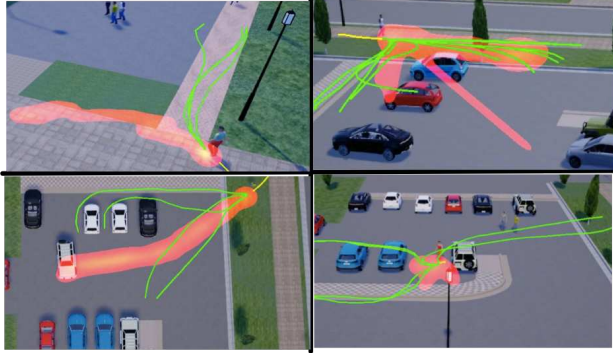


Figure 5: Error analysis. See text for details.

Method	Trained on Real.	Trained on Sim.
Linear	32.19 / 60.92	48.65 / 90.84
LSTM	23.98 / 44.97	28.45 / 53.01
Social-LSTM [1]	23.10 / 44.27	26.72 / 51.26
Social-GAN (V) [15]	30.40 / 61.93	36.74 / 73.22
Social-GAN (PV) [15]	30.42 / 60.70	36.48 / 72.72
Next [30]	19.78 / 42.43	27.38 / 62.11
Ours	18.51 / 35.84	22.94 / 43.35

Table 3: Comparison of different methods on the VI-RAT/ActEV dataset. We report ADE/FDE metrics. First column is for models trained on real video training set and second column is for models trained on the simulated version of this dataset.

ble 4, where the ADE/FDE metrics are shown in the “single-future” column and minADE₂₀/minFDE₂₀ metrics (averaged across all views) in the “multi-future” column. We verify three of our key designs by leaving the module out from the full model.

(1) *Spatial Graph*: Our model is built on top of a spatial 2D graph that uses graph attention to model the scene features. We train model without the spatial graph. As we see, the performance drops on both tasks. (2) *Fine location decoder*: We test our model without the fine location decoder and only use the grid center as the coordinate output. As we see, the significant performance drops on both tasks verify the efficacy of this new module proposed in our study. (3)

Method	Single-Future	Multi-Future
Our full model	18.51 / 35.84	166.1 / 329.5
No spatial graph	28.68 / 49.87	184.5 / 363.2
No fine location decoder	53.62 / 83.57	232.1 / 468.6
No multi-scale grid	21.09 / 38.45	171.0 / 344.4

Table 4: Performance on ablated versions of our model on single and multi-future trajectory prediction. Lower numbers are better.

Multi-scale grid: As mentioned in Section 3, we utilize two different grid scales (36×18) and (18×9) in training. We see that performance is slightly worse if we only use the fine scale (36×18).

6. Conclusion

In this paper, we have introduced the Forking Paths dataset, and the *Multiverse* model for multi-future forecasting. Our study is the first to provide a quantitative benchmark and evaluation methodology for multi-future trajectory prediction by using human annotators to create a variety of trajectory continuations under the identical past. Our model utilizes multi-scale location decoders with graph attention model to predict multiple future locations. We have shown that our method achieves state-of-the-art performance on two challenging benchmarks: the large-scale real video dataset and our proposed multi-future trajectory dataset. We believe our dataset, together with our models, will facilitate future research and applications on multi-future prediction.

Acknowledgements This research was supported by NSF grant IIS-1650994, the financial assistance award 60NANB17D156 from NIST and a Baidu Scholarship. This work was also supported by IARPA via DOI/IBC contract number D17PC00340. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, NIST, DOI/IBC, the National Science Foundation, Baidu, or the U.S. Government.

References

- [1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 1, 2, 5, 6, 7, 8
- [2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *CVPRW*, 2019. 2
- [3] George Awad, Asad Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Joao Magalhaes, David Semedo, and Saverio Blasi. Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *TRECVID*, 2018. 1, 2, 4, 5, 7
- [4] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 1, 2
- [5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 2, 4
- [6] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 1, 2, 4, 5, 6
- [7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 2
- [8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 3, 6
- [9] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPRW*, 2018. 2
- [10] César Roberto de Souza, Adrien Gaidon, Yohann Cabon, and Antonio Manuel López. Procedural generation of videos to train deep action recognition networks. In *CVPR*, 2017. 2
- [11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017. 1, 2, 4
- [12] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 2, 4
- [13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2
- [15] Agrim Gupta, Justin Johnson, Silvio Savarese, Li Fei-Fei, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 1, 2, 4, 5, 6, 7, 8
- [16] Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017. 2
- [17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 6
- [18] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *CVPR*, 2019. 2, 5
- [19] RE Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, D*, 82:35–44, 1960. 1
- [20] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012. 1, 2
- [21] Julian Francisco Pieter Kooij, Nicolas Schneider, Fabian Flohr, and Dariu M Gavrilă. Context-based pedestrian path prediction. In *ECCV*, 2014. 2
- [22] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 3
- [23] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017. 1, 2, 5
- [24] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer Graphics Forum*, pages 655–664. Wiley Online Library, 2007. 2, 4

- [25] Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*, 2016. 4, 6
- [26] Yuke Li. Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In *CVPR*, 2019. 1, 2, 5
- [27] Junwei Liang, Desai Fan, Han Lu, Poyao Huang, Jia Chen, Lu Jiang, and Alexander Hauptmann. An event reconstruction tool for conflict monitoring using social media. In *AAAI*, 2017. 2
- [28] Junwei Liang, Lu Jiang, Liangliang Cao, Yannis Kalantidis, Li-Jia Li, and Alexander G Hauptmann. Focal visual-text attention for memex question answering. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1893–1908, 2019. 2
- [29] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. Focal visual-text attention for visual question answering. In *CVPR*, 2018. 2
- [30] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *CVPR*, 2019. 1, 2, 5, 6, 7, 8
- [31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3
- [32] Matthias Luber, Johannes A Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *ICRA*, 2010. 2
- [33] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *CVPR*, 2017. 2
- [34] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multi-modal future prediction. In *CVPR*, 2019. 1, 2, 4, 6
- [35] Huynh Manh and Gita Alaghband. Scene-1stm: A model for human trajectory prediction. *arXiv preprint arXiv:1808.04018*, 2018. 2
- [36] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 1, 2, 4, 5, 7
- [37] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2012. 4
- [38] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018. 6
- [39] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *ACM Multimedia*, 2017. 2
- [40] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. 4
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 4
- [42] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *ICCV*, 2017. 1
- [43] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *ECCV*, 2018. 1, 2, 4, 6
- [44] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. *arXiv preprint arXiv:1905.01296*, 2019. 2, 4, 5, 6
- [45] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2
- [46] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016. 4
- [47] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 2, 4
- [48] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*, 2017. 1
- [49] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018. 2, 5, 6
- [50] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *ECCV*, 2018. 2

- [51] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018. 2
- [52] Chen Sun, Per Karlsson, Jiajun Wu, Joshua B Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *arXiv preprint arXiv:1902.09641*, 2019. 2
- [53] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. *arXiv preprint arXiv:1911.00997*, 2019. 1, 2
- [54] Luca Anthony Thiede and Pratik Prabhanjan Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. *arXiv preprint arXiv:1907.10178*, 2019. 1, 2, 4
- [55] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3
- [56] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *ICLR*, 2019. 3
- [57] Yu Wu, Lu Jiang, and Yi Yang. Revisiting embodiedqa: A simple baseline and beyond. *arXiv preprint arXiv:1904.04166*, 2019. 2
- [58] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015. 2, 3, 6
- [59] Hao Xue, Du Q Huynh, and Mark Reynolds. Sslstm: A hierarchical lstm model for pedestrian trajectory prediction. In *WACV*, 2018. 2
- [60] Takuma Yagi, Kartikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *CVPR*, 2018. 2
- [61] Matthew D Zeiler. Adadelata: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 6
- [62] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *CVPR*, 2019. 2, 5
- [63] Yiwei Zhang, Graham M Gibson, Rebecca Hay, Richard W Bowman, Miles J Padgett, and Matthew P Edgar. A fast 3d reconstruction system with a low-cost camera accessory. *Scientific reports*, 5:10909, 2015. 2
- [64] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *CVPR*, 2019. 2, 5
- [65] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 3, 6
- [66] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017. 2