SimAug: Learning Robust Representations from 3D Simulation for Pedestrian Trajectory Prediction in Unseen Cameras

Junwei Liang[1], Lu Jiang[2], and Alexander Hauptmann[1]

[1] Carnegie Mellon University
{junweil,alex}@cs.cmu.edu
[2] Google Research
lujiang@google.com

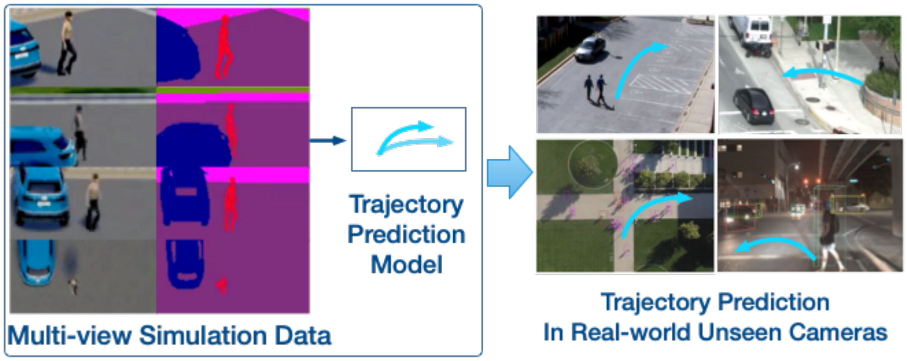
Trajectory Prediction Model
Multi-view Simulation Data
Trajectory Prediction
In Real-world Unseen Cameras

Fig. 1. Illustration of pedestrian trajectory prediction in unseen cameras. We propose to learn robust representations only from 3D simulation data that could generalize to real-world videos captured by unseen cameras.

Abstract. This paper focuses on the problem of predicting future trajectories of people in unseen scenarios and camera views. We propose a method to efficiently utilize multi-view 3D simulation data for training. Our approach finds the hardest camera view to mix up with adversarial data from the original camera view in training, thus enabling the model to learn robust representations that can generalize to unseen camera views. We refer to our method as *SimAug*. We show that *SimAug* achieves best results on three out-of-domain real-world benchmarks, as well as getting state-of-the-art in the Stanford Drone and the VIRAT/ActEV dataset with in-domain training data. We will release our models and code.

Keywords: Trajectory Prediction, 3D Simulation, Synthetic Data, Representation Learning, Adversarial Learning

1 Introduction

Future trajectory prediction [24,1,19,34,53,28,33] is a fundamental problem in video analytics, which aims at forecasting a pedestrian's future path in video in the next few seconds. Recent advancements in future trajectory prediction have been promising in a variety of vision applications such as self-driving vehicles [4,7,8], safety monitoring [34], robotic planning [47,48], etc.

A notable bottleneck for existing works is that the current model is closely coupled with the video cameras on which it is trained, and generalizes poorly on new cameras with novel views or scenes. For example, prior works have proposed various models to forecast a pedestrian's trajectories in video cameras of different types such as stationary outdoor cameras [44,32,1,19,29,37], drone cameras [53,13,30], ground-level egocentric cameras [70,47,58], or dash cameras [42,57,8]. Models that rely on RGB pixel inputs are especially vulnerable to view changes [34,33,53]. Previous works [1,19,53] have tried to conduct leave-one-scene-out experiments on ETH/UCY [45,29] dataset under similar camera views. However, existing models are all separately trained and tested within one or two datasets, and there have been no attempts at successfully generalizing the model to other datasets of novel camera views. This bottleneck significantly hinders the application whenever there is a new camera because it requires annotating new data to fine-tune the model. Since it takes time to record data in new cameras, such annotation procedure is not only expensive but also can considerably delay the deployment of the model.

An ideal model should be able to disentangle human behavioral dynamics from specific camera views, and produces robust future prediction despite variances in camera views. Motivated by this idea, in this work, we learn a robust model for future trajectory prediction that can generalize to unseen video cameras. Different from existing works, our goal is to train a model only on synthetic data but test the model, out of the box, on unseen real-world videos, without further re-training or fine-tuning. Following the success of learning from simulation [52,56,64,76,15,49], we utilize a 3D simulator CARLA [14] to reconstruct static scene and dynamic elements from the VIRAT/ActEV videos [44], and generate multiple views and pixel-precise semantic segmentation labels for training. As illustrated in Figure 1, examples of pedestrian trajectory are recorded in multiple camera views to facilitate robust training. Meanwhile, scene semantic segmentation is used instead of RGB pixels to help alleviate the influence of different lighting conditions, different texture of the scene and subtle noises produced by camera sensors, etc. Both designs help better model transfer to unseen real-world cameras.

As unseen real-world videos are not available during training in our problem, traditional approaches for domain adaptation [5,16,62] or learning using privileged information [27,63,36] may not be applicable. To tackle this issue, we propose a new data augmentation method called *SimAug* to augment the representation such that it is robust to the variances in semantic scenes and camera views. First, to deal with the gap between real and synthetic semantic scene, we represent each training trajectory by high-level scene semantic segmentation features, and defend our model from adversarial examples generated by white-box attack methods [18]. Second, to overcome the changes in camera views, we generate multiple views for the same trajectory, and encourage the model to focus on the "hardest" view to which the model has learned. Following [22,35], the classification loss is adopted and the view with the highest loss is favored during training. Finally, the augmented trajectory is computed as a convex combination of the trajectories generated in previous steps. Our trajectory prediction model is built on a multi-scale representation in [33] and the final model is trained to minimize the empirical vicinal risk over the distribution of augmented trajectories. Our method is partially inspired by robust deep learning methods using adversarial training [40,61,67,10], Mixup [74], and MentorNet [22].

We empirically validate our model, which is trained only on simulation data, on three real-world benchmarks for future trajectory prediction: VIRAT/ActEV [44,2], Stanford Drone [50] and Argoverse [8]. These three benchmarks represent three different camera views: 45-degree view, top-down view and dashboard camera view with ego-motions. The results show our method performs favorably against baseline methods including standard augmentation, adversarial learning and imitation learning. Notably, our method achieves better results to the state-of-the-art on the VIRAT/ActEV and Stanford Drone benchmark. We will release our data, models and code. To summarize, our contribution is threefold:

- We study a new setting of future trajectory prediction: the model is trained once and used on any unseen video camera.
- We propose a new and effective approach to augment the representation of trajectory prediction models using multi-view simulation data.
- Ours is the first work on future trajectory prediction to demonstrate the efficacy of training on 3D simulation data, and establish new state-of-the-art results on three public benchmarks.

2 Related Work

Trajectory prediction. Recently there are a large body of work on predicting person future trajectories in a variety of scenarios. Many works [1,43,69,75,34,33,53] focused on modeling person motions in videos recorded with stationary cameras. Datasets like VIRAT/ActEV [44], ETH/UCY [29,37] and Stanford Drone [50] have been used for evaluating pedestrian trajectory prediction. Social-LSTM [1] added social pooling to model nearby pedestrian trajectory patterns. Social-GAN [19] added adversarial network [17] on Social-LSTM to generate diverse future trajectories. Several works focused on learning the effects of the physical scene, e.g., people tend to walk on the sidewalk instead of grass. Kitani et al. in [24] used Inverse Reinforcement Learning to forecast human trajectory. SoPhie [53] combined deep neural network features from scene semantic segmentation model and generative adversarial network (GAN) using attention to model person trajectory. Some recent works [25,70,39,34] have attempted to predict person paths by utilizing individuals' visual features instead of considering them as points in the scene. Recently Liang et al. [33] proposed to use abstract scene semantic segmentation features for better generalization. Many works [28,54,4,21,78,41,30,48] have been proposed for top-down view videos for trajectory prediction. Notably, the Stanford Drone Dataset (SDD) [50] is used in many works [53,13,30] for trajectory prediction with drone videos. Other works have also looked into pedestrian prediction in dashcam videos [42,57,25,28] and first-person videos [70,58]. Many vehicle trajectory datasets [6,8,71] have been proposed as a result of self-driving's surging popularity.

Learning from 3D simulation data. As the increasing research focus in 3D computer vision [77,31,55,14,49,51,20], many research works [46,15,56,12,66,80,59,33,60,3] have used 3D simulation for training and evaluating real-world tasks. Many works [46,15,56] were proposed to use data generated from 3D simulation for video object detection, tracking and action recognition analysis. Sun et al. [59] proposed a forecasting model by using a gaming simulator. AirSim [55] and CARLA [14] were proposed for robotic autonomous controls for drones and vehicles. Zeng et al. [73] proposed to use 3D simulation for adversarial attacks. RSA [76] used randomized simulation data for human action recognition. The ForkingPaths dataset [33] was proposed for evaluating multi-future trajectory prediction. Human annotators were asked to control agents in a 3D simulator to create a multi-future trajectory dataset. Different from the ForkingPaths dataset, we propose an efficient method to reconstruct trajectories in real-world videos and collecting large-scale multi-view data.

Data augmentation and adversarial learning. Traditional domain adaptation approaches [5,16,62,23], including unsupervised domain adaptation, may not be applicable as our target domain is considered "unseen" during training. Methods for learning using privileged information (LUPI) [27,63,36,38] is not applicable for a similar reason. Closest to ours is robust deep learning methods. Recent works focused on the following directions: (i) *adversarial training* [18,40,67,73] defends the adversarial attacks generated on-the-fly during training using gradient-based methods [40,18,61]; (ii) data augmentation methods to overcome unknown variances between training and text examples such as *Mixup* [74], *AutoAgument* [11], etc. Different from prior work, ours uses 3D simulation data as a new way of doing data augmentation. Also, our proposed augmentation method is carefully designed for future trajectory prediction. The key design choices are substantiated by our ablation studies.

3 Approach

In this section, we describe our approach to learn robust models for future trajectory prediction, which we call *SimAug*. Our goal is to train a model only on simulation training data that can effectively predict the future trajectory in real-world test videos that are unseen during training.

3.1 Problem Formulation

We focus on predicting the locations of a single agent for multiple steps into the future. Given a sequence of historic video frames $V_{1:h}$ of the past h steps and the

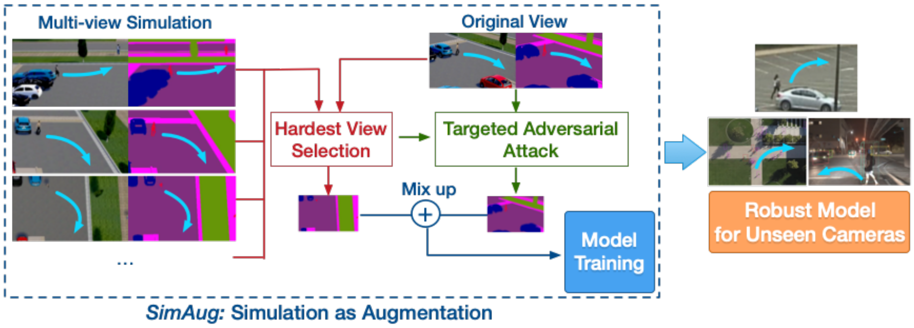
Multi-view Simulation
Original View
Hardest View Selection
Targeted Adversarial Attack
Mix up
Model Training
Robust Model for Unseen Cameras
SimAug: Simulation as Augmentation

Fig. 2. Overview of our method. We efficiently generate multi-view trajectory data as described in Section 3.2. At each training iteration, the hardest camera view of the same trajectory sample is selected into training along with the adversarial sample of the original view.

past agent locations $L_{1:h}$ in training, we learn a probabilistic model on simulation data to estimate $P(L_{h+1:T}|L_{1:h}, V_{1:h})$ for $T-h$ steps into the future. At test time, our model takes as input an agent's observable past $(V_{1:h}, L_{1:h})$ in real videos to predict the agent's future locations $L_{h+1:T} = \{y_{h+1}, \ldots, y_T\}$, where y_t is the location coordinates. As the test videos are unseen during training, the model is supposed to be invariant to the variances in semantic scenes, camera views, and camera motions.

3.2 Training Data Generation From Simulation

Our model is trained only on simulation data. To ensure high-level realism, the training trajectories are generated by CARLA [14], an open source 3D simulator built on top of the state-of-the-art game engine *Unreal Engine 4*. We use the trajectories from the Forking Paths dataset [33] that are semi-manually recreated from the VIRAT/ActEV benchmark by projecting real-world annotations to the 3D simulation world. Note that it is not our intention to build an exact replica of the real-world scene, nor it is necessary to help train a model for real-world task as suggested in previous works [15,51,33,76].

With CARLA, we record multiple views of the same trajectory of different camera angles and positions. For a trajectory $(V_{1:T}, L_{1:T})$ in original view, let $\mathcal{S} = \{(V^{(i)}_{1:T}, L^{(i)}_{1:T})\}^{|\mathcal{S}|}_{i=1}$ denote a set of additional views for the same trajectory. In our experiments, we use four camera parameters pre-specified in [33], i.e. three 45-degree views and one top-down view. We use a total of 4 scenes. See Fig. 3. The ground-truth location varies under different camera views and $L^{(i)}_{1:T} \neq L^{(j)}_{1:T}$ for $i \neq j$. Note that these camera positions and angles are defined in [33] specifically for VIRAT/ActEV dataset. The top-down view cameras in Stanford Drone dataset [50] are still considered unseen to the model since the scenes and camera positions are quite different.

In simulation, we also collect ground truth scene semantic segmentation from $K = 13$ classes including sidewalk, road, vehicle, pedestrian, etc. At test time, we

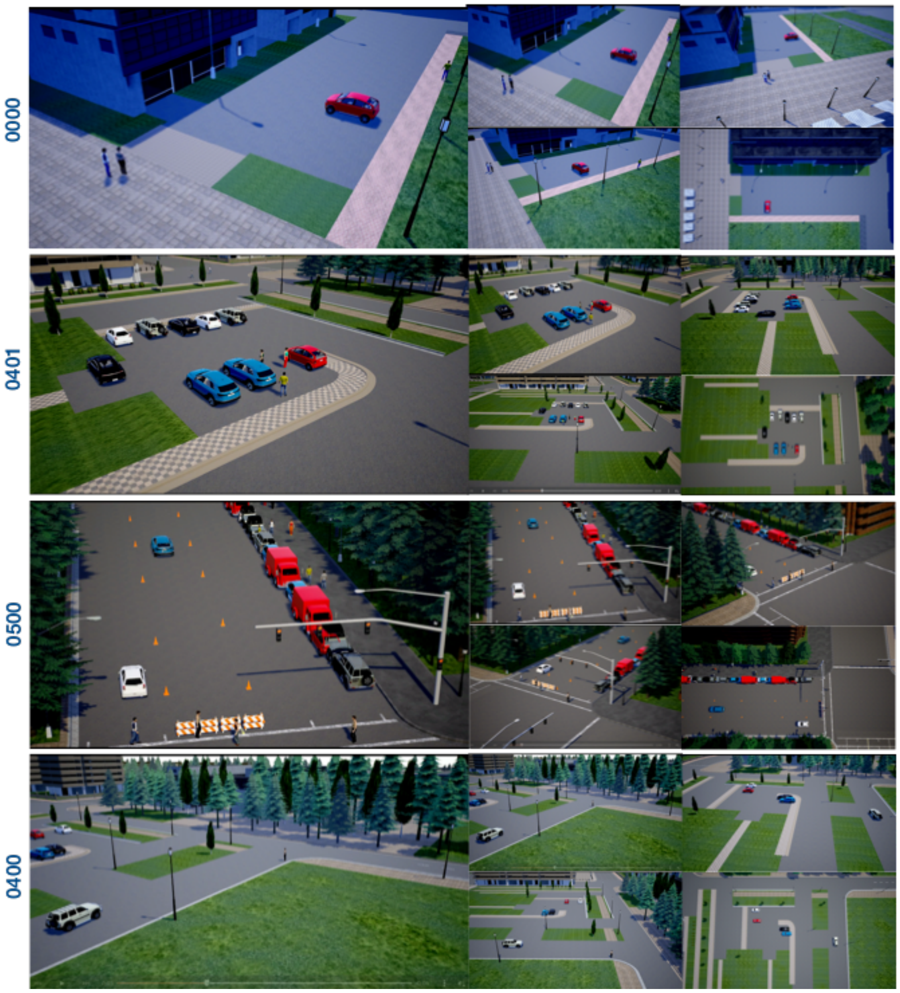

Fig. 3. Visualization of the multi-view 3D simulation data used in *SimAug* training. Data generation process is described in Section 3.2. We use 4 camera views from 4 scenes defined in [33]. “0400” and “0401” scene have overlapping views. The views in the left column are the original views from VIRAT/ActEV dataset.

extract the semantic segmentation feature using a pre-trained model with same number of class labels per pixel. To be more specific, we use the Deeplab model [9] trained on the ADE20k [79] dataset and keep its weights frozen. To bridge the gap between real and simulated video frames, we represent all trajectory $V_{1:T}$ as a sequence of scene semantic segmentation features.

3.3 Multi-view Simulation Augmentation (*SimAug*)

In this subsection, we describe *SimAug* for learning robust models. Given a trajectory in its original view $(V_{1:T}, L_{1:T})$, we generate a set of additional views in $\mathcal{S} = \{(V_{1:T}^{(i)}, L_{1:T}^{(i)})\}_{i=1}^{|\mathcal{S}|}$ as described in the previous section, where $V_t^{(i)}$ represents scene semantic features of view i at time t. $L_{1:T}^{(i)}$ is a sequence of ground-truth locations for the i-th view.

We use the original view as an anchor to search for the "hardest" view that is most inconsistent with what the model has learned. Inspired by [22], we use the classification loss as the criteria and compute:

$$j^* = \underset{j \in [1,|S|]}{\operatorname{argmax}} \{\mathcal{L}_{\text{cls}}(V_{1:h} + \delta, L_{h+1:T}^{(j)}; \theta)\}, \tag{1}$$

where δ is an ℓ_∞-bounded random perturbation applied to the input features. $\mathcal{L}_{\text{cls}}$ is the location classification loss which will be discussed in the next subsection.

Then for the original view, we generate an adversarial trajectory feature by the targeted-FGSM attack [26]:

$$V_{1:h}^{adv} = V_{1:h} - \epsilon \cdot \operatorname{sign}(\nabla_{V_{1:h}} \mathcal{L}_{\text{cls}}(V_{1:h} + \delta, L_{h+1:T}^{(j^*)}; \theta)), \tag{2}$$

where ϵ is the hyper-parameter to be chosen. The attack forces the adversarial trajectory to be predicted as the true future locations in the selected camera view. In essence, the resulting adversarial feature is "warped" to the "hardest" camera view by a small perturbation. By defending against such adversarial examples, our model learns representations that are robust against changes in camera views.

Finally, we mix up the trajectory of the selected view and the adversarial trajectory by a convex combination function over their features and one-hot location labels.

$$\begin{aligned} V_{1:h}^{aug} &= \lambda \cdot V_{1:h}^{adv} + (1-\lambda) \cdot V_{1:h}^{(j^*)} \\ y_t^{aug} &= \lambda \cdot \text{one-hot}(y_t) + (1-\lambda) \cdot \text{one-hot}(y_t^{(j^*)}) \quad t \in [h+1, T] \end{aligned} \tag{3}$$

where $L_{h+1:T}^{aug} = \{y_{h+1}^{aug}, \ldots, y_T^{aug}\}$ and the one-hot$(\cdot)$ function projects xy coordinates into an one-hot embedding over a predefined grid used in computing the classification loss. Following [74], λ is drawn from a Beta distribution Beta(α, α) controlled by the hyper-parameter α.

The detailed algorithm for training with one training step is listed in Algorithm 1. To train robust models to various camera views and semantic scenes, we learn representations over augmented training examples to overcome (i) random feature perturbations (ii) targeted adversarial attack, and (iii) the "hardest" feature from other views. By the mix-up step in Eq. (3), our model is trained to minimize empirical vicinal risk over a new distribution constituted by the generated augmented trajectories, which is proved to be useful in improving model robustness in CNN training [74].

Algorithm 1: Multi-view Simulation Adversarial Augmentation (*SimAug*)

```
Input  : Mini-batch of trajectories; hyper-parameters $\alpha$ and $\epsilon$
Output: Classification loss $\mathcal{L}_{\text{cls}}$ computed over augmented trajectories
1 for each trajectory $(V_{1:T}, L_{1:T})$ in the mini-batch do
2     Generate trajectories from additional views $\mathcal{S} = \{(V^{(i)}_{1:T}, L^{(i)}_{1:T})\}$;
3     Compute the loss for each camera view using $\mathcal{L}_{\text{cls}}(V_{1:h} + \delta, L^{(j)}_{h+1:T}; \theta)$;
4     Select the view with the largest loss $j^*$ by Eq. (1) ;
5     Generate an adversarial trajectory $V^{adv}_{1:h}$ by Eq. (2);
6     Mix up $(V^{adv}_{1:h}, L_{h+1:T})$ and $(V^{(j^*)}_{1:h}, L^{(j*)}_{h+1:T})$ by Eq. (3);
7     Compute $\mathcal{L}_{\text{cls}}$ over the augmented trajectory $(V^{aug}_{1:h}, L^{aug}_{h+1:T})$ from Step 6;
8 end
9 return averaged $\mathcal{L}_{\text{cls}}$ over the augmented trajectories
```

3.4 Trajectory Prediction Model

We build our backbone on Multiverse [33], a state-of-the-art multi-future trajectory prediction model. We use *SimAug* to improve the robustness of Multiverse view-invariant representation. Our method is general and flexible to be used in other models.

Input Features. The model is given the past locations, $L_{1:h}$, and the scene, $V_{1:h}$. Each ground truth location L_t is encoded by an one-hot vector $y_t \in \mathbb{R}^{HW}$ representing the nearest cell in a 2D grid of size $H \times W$. In our experiment, we use a grid scale of 36×18. Each video frame V_t is encoded to as semantic segmentation feature of size $H \times W \times K$ where $K = 13$ is the total number of class labels. As discussed in previous section, we use *SimAug* to generate augmented trajectories $(V^{aug}_{1:h}, L^{aug}_{1:h})$ as our input during training.

History Encoder. A convolutional RNN [68,65] is used to get the final spatial-temporal feature state $H_t \in \mathbb{R}^{H \times W \times d_{enc}}$, where d_{enc} is the hidden size. The context is represented as the last hidden state and the history video frames, $\mathcal{H} = [H_h, V_{1:h}]$.

Location Decoder. After getting the context $\mathcal{H}$, a coarse location decoder is used to predict locations at the level of grid cells at each time-instant by:

$$\hat{y}_t = \text{softmax}(f_c(\mathcal{H}, H^c_{t-1})) \in \mathbb{R}^{HW} \tag{4}$$

where f_c is the convolutional recurrent neural network (ConvRNN) with graph attention proposed in [33] and H^c_t is the hidden state of the ConvRNN. Then a fine location decoder is used 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 by:

$$\hat{O}_t = \text{MLP}(f_o(\mathcal{H}, H^o_{t-1})) \in \mathbb{R}^{HW\times 2}, \tag{5}$$

where f_o is a separate ConvRNN and H^o_t is its hidden state. To compute the final prediction location, we use

$$\hat{L}_t = Q_g + \hat{O}_{tg} \tag{6}$$

where $g = \text{argmax}\,\hat{y}_t$ is the index of the selected grid cell, $Q_g \in \mathbb{R}^2$ is the center of that cell, and $\hat{O}_{tg} \in \mathbb{R}^2$ is the predicted offset for that cell at time t.

Training. We use *SimAug* (see Section 3.3) to generate $L^{aug}_{h+1:T} = \{y^{aug}_{h+1}, \ldots, y^{aug}_T\}$ as labels for training. For the coarse decoder, the cross-entropy loss is used:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{T}\sum_{t=h+1}^{T}\sum_{c=1}^{HW} y^{aug}_{tc}\log(\hat{y}_{tc}) \tag{7}$$

For the fine decoder, we use the original ground truth label $L_{h+1:T}$:

$$\mathcal{L}_{\text{reg}} = \frac{1}{T}\sum_{t=h+1}^{T}\sum_{c=1}^{HW} \text{smooth}_{l_1}(O_{tc}, \hat{O}_{tc}) \tag{8}$$

where $O_{tc} = L_t - Q_c$ is the delta between the ground true location and the center of the c^{th} grid cell. The final loss is then calculated using

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

where λ_2 controls the ℓ_2 regularization (weight decay), and $\lambda_1 = 0.5$ is used to balance the regression and classification losses.

4 Experiments

In this section, we evaluate various methods, including our *SimAug* method, on three public video benchmarks of real-world videos captured under different camera views: the VIRAT/ActEV [2,44] dataset, the Stanford Drone dataset [50], and the autonomous driving dataset Argoverse [8]. We demonstrate the efficacy of our method for unseen cameras in Section 4.2 and how our method can also improve state-of-the-art when fine-tuned on the real training data in Section 4.3 and Section 4.4.

4.1 Evaluation Metrics

Let $L^i = L^i_{t=(h+1)\cdots T}$ be the true future trajectory for the i^{th} test sample, and $\hat{L}^{ik}$ be the corresponding k^{th} prediction sample, for $k \in [1, K]$.

i) *Minimum Average Displacement Error Given K Predictions* ($\mathrm{minADE_K}$): for each true trajectory sample i, we select the closest K predictions, and then measure its average error:

$$\mathrm{minADE}_K = \frac{\sum_{i=1}^{N} \min_{k=1}^{K} \sum_{t=h+1}^{T} \|L_t^i - \hat{L}_t^{ik}\|_2}{N \times (T-h)} \tag{10}$$

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

$$\mathrm{minFDE}_K = \frac{\sum_{i=1}^{N} \min_{k=1}^{K} \|L_T^i - \hat{L}_T^{ik}\|_2}{N} \tag{11}$$

iii) *Grid Prediction Accuracy* (Grid_Acc): As our base model also predicts coarse grid locations as described in Section 3.4, we also evaluate the accuracy between the predicted grid $\hat{y}_t$ and the ground truth grid y_t.

4.2 Main Results

Dataset & Setups. In the following experiments, we compare *SimAug* with classical data augmentation methods as well as recent adversarial learning methods to train robust representations. All methods are trained using the same backbone on the same **simulation training data** described in Section 3.2, and tested on three public benchmarks. All real videos are not allowed to be used during training except in our finetuning experiments. For VIRAT/ActEV, we use the same test split as [34,33]. For SDD, we utilize the standard test split as [53,13] and for Argoverse, we use the official validation set from the 3D tracking task as our test set. The videos from the "ring_front_center" camera are used.

These datasets have different levels of difficulties. VIRAT/ActEV is the easiest one as we have used its training trajectories projected in our simulation training data. SDD is more difficult as its camera positions and scenes are different from our training. Argoverse is the most challenging one with distinct scenes, camera views, and ego-motions.

Following the setting in previous works [34,1,19,1,19,53,41,33,13], the models observe 3.2 seconds (8 frames) of every pedestrian and predict the future 4.8 seconds (12 frames) of person trajectory. We use the pixel values for the trajectory coordinates as it is done in [70,34,28,7,30,41,4,21,13].

Baseline methods. We compare *SimAug* with the following baseline methods for learning robust representations. All methods are built on the base model and trained using the same simulation training data. ***Base Model*** is the trajectory prediction model proposed in [33]. ***Standard Aug*** is the base model trained with standard data augmentation techniques including horizontal flipping and random input jittering. ***Fast Gradient Sign Method (FGSM)*** is the base model trained with adversarial examples generated by the targeted-FGSM attack method [18]. We use random labels for the targeted-FGSM attack.

Projected Gradient Descent (PGD) is learned with a recent iterative adversarial learning method [40,67]. The number of iteration is set to 10 and other hyper-parameters follow [67].

Implementation Details. We follow the implementation in [33] and use it as our base model. To be more specific, we use $\alpha = 0.2$ for the Beta distribution in Eq (3) and we use $\epsilon = \delta = 0.1$ in Eq (2). We use a total of 4 camera views in training, including three 45-degree views and one top-down view. See Section 3.2. All models are trained using Adadelta optimizer [72] with an initial learning rate of 0.3 and a weight decay of 0.001. Other hyper-parameters for the baselines are the same as the ones in [33]. We evaluate the top $K = 1$ future trajectory prediction of all models.

Quantitative Results. Table 1 shows the evaluation results. As we see, our method performs favorably against other baseline methods across all three evaluation metrics and all three benchmarks. In particular, "Standard Aug" seems to be not generalizing well to unseen cameras. FGSM improves significantly on the "Grid_Acc" metric but fails to translate the improvement to final location predictions. *SimAug* is able to improve the model overall due to the effective use of multi-view data. All other methods are unable to improve trajectory prediction on Argoverse, whose data characteristics include ego-motions and distinct dashboard-view cameras. The results substantiate the efficacy of *SimAug* for trajectory prediction in unseen cameras.

Qualitative Analysis. We visualize outputs of our base model with and without *SimAug* in Fig. 4. We show visualizations on all three datasets. In each image, the yellow trajectories are history trajectories and the green ones are ground truth future trajectories. Outputs of the base model without SimAug are colored with blue heatmaps and the yellow-orange heatmaps are from the same model with SimAug. As we see, the base model with SimAug augmentation yields more accurate trajectories with turnings (Fig. 4 1a., 3a.) while without it the model sometimes predicts the wrong turns (Fig. 4 1b., 1c., 1d., 2a., 3a., 3c., 3d.). In addition, the length of SimAug model predictions is more accurate (Fig. 4 2b., 2c., 2d., 3b.).

Table 1. Comparison to standard data augmentation method and recent adversarial learning methods on three datasets. We report Grid_Acc(↑)/minADE$_1$(↓)/minFDE$_1$(↓) metrics. All methods are built on the backbone model in [33] and trained using the same multi-view simulation data described in Section 3.2.

Method	VIRAT/ActEV	Stanford Drone	Argoverse
Base Model [33]	44.2%/26.2/49.7	31.4%/21.9/42.8	26.6%/69.1/183.9
Standard Aug	45.5%/25.8/48.3	21.3%/23.7/47.6	28.9%/70.9/183.4
PGD [40,67]	47.5%/25.1/48.4	28.5%/21.0/42.2	25.9%/72.8/184.0
FGSM [18]	48.6%/25.4/49.3	42.3%/19.3/39.9	29.2%/71.1/185.4
SimAug	**51.1%/21.7/42.2**	**45.4%/15.7/30.2**	**30.9%/67.9/175.6**

4.3 State-of-the-Art Comparison on Stanford Drone Dataset

In this section, we compare our *SimAug* model with the state-of-the-art generative models, including Social-LSTM, Social-GAN, DESIRE, and SoPhie [1,19,53,28]. We also compare with imitation learning model, IDL [30], and inverse reinforcement learning model, P2T$_{\text{IRL}}$ [13] for trajectory prediction on the Stanford Drone Dataset. Following previous works, we evaluate our method with minimal errors over 20 predictions.

Results & Analysis. The results are shown in Table 2, where *SimAug* is built on top of the *Multiverse* model. As it shows, *SimAug* model trained only on out-domain simulation data (second to the last row) achieves comparable or even better performance than other state-of-the-art models that are trained on in-domain real videos. By further fine-tuning on the learned representations of *SimAug*, we achieve the state-of-the-art performance on the Stanford Drone Dataset. The promising results demonstrate the efficacy of *SimAug* for trajectory prediction in unseen cameras.

Table 2. State-of-the-art comparison on the Stanford Drone Dataset (SDD). Numbers are minimal errors over 20 predictions and lower the better. Baseline numbers are taken from [53,13]. "SimAug" is trained without using SDD training data and "SimAug + finetune" is further finetuned on SDD training data.

Method	minADE$_{20}$(↓)	minFDE$_{20}$ (↓)
Social-LSTM [1]	31.19	56.97
Social-GAN [19]	27.25	41.44
DESIRE [28]	19.25	34.05
SoPhie [53]	16.27	29.38
Multiverse [33]	14.78	27.09
IDL [30]	13.93	24.40
P2T$_{\text{IRL}}$ [13]	12.58	22.07
SimAug	12.03	23.98
SimAug + finetune	**10.27**	**19.71**

4.4 State-of-the-Art Comparison on VIRAT/ActEV

In this section, we compare our *SimAug* model with state-of-the-art models on VIRAT/ActEV. Following the previous work [33], we evaluate our method with errors in the top 1 prediction. Experimental results are shown in Table 3, where all models in the top rows are trained on the real-world training videos in VIRAT/ActEV. Our model trained on simulation data achieves competitive performance and with fine-tuning we achieve the best performance on the VIRAT/ActEV benchmark.

4.5 Ablation Experiments

We test various ablations of our approach to validate our design decisions. Results are shown in Table 4, where the top 1 prediction is used in evaluations. We verify four key design choices by removing each, at a time, from the full model.

Table 3. State-of-the-art comparison on the VIRAT/ActEV dataset. Numbers are minimal errors over 1 predictions and lower the better.

Method	minADE$_1$(↓)	minFDE$_1$ (↓)
Social-LSTM [1]	23.10	44.27
Social-GAN [19]	30.42	60.70
Next [34]	19.78	42.43
Multiverse [33]	18.51	35.84
SimAug	21.73	42.22
SimAug + finetune	**17.96**	**34.68**

(1) *Multi-view data:* Our method is trained on multi-view simulation data and we use 4 camera views in our experiments. We test our method without one of the camera view (top-down view) that is similar to the ones in SDD dataset to see the effects. As we see, the performance drops due to fewer number of data and less diverse views, suggesting that we should use more views in *SimAug* (which is effortless to do in 3D simulator).

(2) *Random perturbation:* We test our model without random perturbation on the original view trajectory samples by setting $\delta = 0$ (Eq. (1)). As we see, performance drops on all three datasets and particularly on the more difficult Argoverse dataset.

(3) *Adversarial attack:* We test our model without adversarial attack by replacing Eq. (2) with $V_{1:h}^{adv} = V_{1:h}$. The performance drops slightly across all three benchmarks.

(4) *View selection:* We replace Eq. (1) with random search to see the effect of view selection. As we see, the significant performance drops on trajectory prediction verify the effectiveness of our design.

Table 4. Performance on ablated versions of our method on three benchmarks. We report Grid_Acc(↑)/minADE$_1$(↓)/minFDE$_1$(↓) metrics.

Method	VIRAT/ActEV	Stanford Drone	Argoverse
SimAug full model	51.1%/21.7/42.2	45.4%/15.7/30.2	30.9%/67.9/175.6
- top-down view data	51.0%/22.8/43.6	44.9%/18.4/35.6	30.9%/68.4/178.3
- random perturbation	49.1%/23.6/43.8	44.2%/18.7/35.6	32.6%/69.1/180.2
- adversarial attack	49.2%/23.1/43.8	43.1%/17.4/32.9	26.4%/68.0/177.5
- view selection	52.0%/23.0/42.9	47.4%/19.6/38.2	32.4%/68.6/177.0

5 Conclusion

In this paper, we have introduced *SimAug*, which utilizes multi-view 3D simulation data to learn robust representations for trajectory prediction. We have shown that our method achieves competitive performance on three public benchmarks with and without using the real-world training data. We believe our approach will facilitate future research and applications on robust future prediction using 3D simulation.

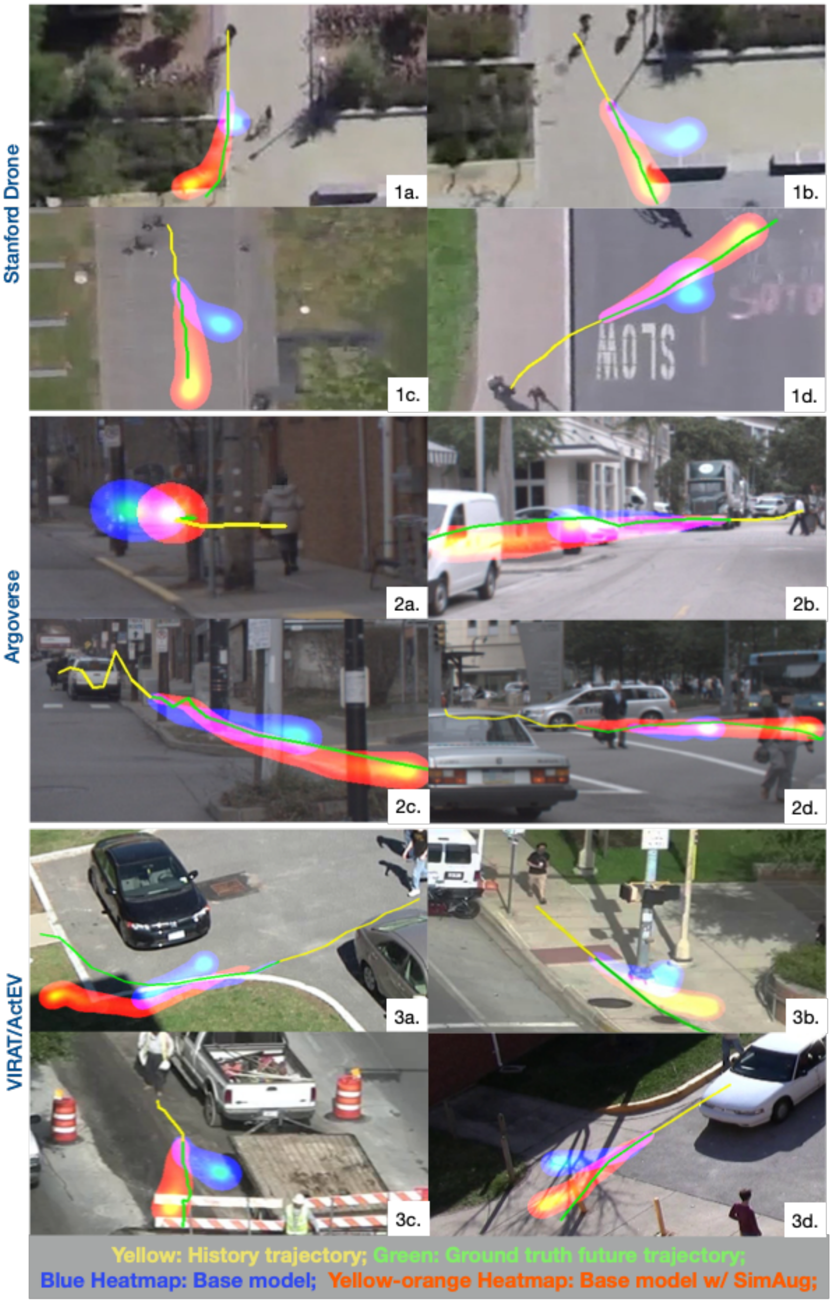
Stanford Drone
Argoverse
VIRAT/ActEV
1a.
1b.
1c.
1d.
2a.
2b.
2c.
2d.
3a.
3b.
3c.
3d.
Yellow: History trajectory; Green: Ground truth future trajectory;
Blue Heatmap: Base model; Yellow-orange Heatmap: Base model w/ SimAug;

Fig. 4. Qualitative analysis. Trajectory prediction from different models are colored and overlaid in the same image. See text for details.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: CVPR (2016)
2. Awad, G., Butt, A., Curtis, K., Fiscus, J., Godil, A., Smeaton, A.F., Graham, Y., Kraaij, W., Qunot, G., Magalhaes, J., Semedo, D., Blasi, S.: Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In: TRECVID (2018)
3. Bak, S., Carr, P., Lalonde, J.F.: Domain adaptation through synthesis for unsupervised person re-identification. In: ECCV (2018)
4. Bansal, M., Krizhevsky, A., Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079 (2018)
5. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR (2017)
6. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
7. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449 (2019)
8. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., et al.: Argoverse: 3d tracking and forecasting with rich maps. In: CVPR (2019)
9. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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)
10. Cheng, Y., Jiang, L., Macherey, W.: Robust neural machine translation with doubly adversarial inputs. ACL (2019)
11. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: CVPR (2019)
12. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: CVPRW (2018)
13. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735 (2020)
14. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938 (2017)
15. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: CVPR (2016)
16. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
19. Gupta, A., Johnson, J., Savarese, Li Fei-Fei, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: CVPR (2018)

20. Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S., Riedmiller, M., et al.: Emergence of locomotion behaviours in rich environments. arXiv preprint arXiv:1707.02286 (2017)
21. Hong, J., Sapp, B., Philbin, J.: Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: CVPR (2019)
22. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. arXiv preprint arXiv:1712.05055 (2017)
23. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR (2019)
24. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: ECCV (2012)
25. Kooij, J.F.P., Schneider, N., Flohr, F., Gavrila, D.M.: Context-based pedestrian path prediction. In: ECCV (2014)
26. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. ICLR (2017)
27. Lambert, J., Sener, O., Savarese, S.: Deep learning under privileged information using heteroscedastic dropout. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8886–8895 (2018)
28. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: CVPR (2017)
29. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum. pp. 655–664. Wiley Online Library (2007)
30. Li, Y.: Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In: CVPR (2019)
31. Liang, J., Fan, D., Lu, H., Huang, P., Chen, J., Jiang, L., Hauptmann, A.: An event reconstruction tool for conflict monitoring using social media. In: AAAI (2017)
32. Liang, J., Jiang, L., Cao, L., Kalantidis, Y., Li, L.J., Hauptmann, A.G.: Focal visual-text attention for memex question answering. IEEE transactions on pattern analysis and machine intelligence **41**(8), 1893–1908 (2019)
33. Liang, J., Jiang, L., Murphy, K., Yu, T., Hauptmann, A.: The garden of forking paths: Towards multi-future trajectory prediction. In: CVPR (2020)
34. Liang, J., Jiang, L., Niebles, J.C., Hauptmann, A.G., Fei-Fei, L.: Peeking into the future: Predicting future person activities and locations in videos. In: CVPR (2019)
35. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
36. Lopez-Paz, D., Bottou, L., Schölkopf, B., Vapnik, V.: Unifying distillation and privileged information. arXiv preprint arXiv:1511.03643 (2015)
37. Luber, M., Stork, J.A., Tipaldi, G.D., Arras, K.O.: People tracking with human motion predictions from social forces. In: ICRA (2010)
38. Luo, Z., Hsieh, J.T., Jiang, L., Carlos Niebles, J., Fei-Fei, L.: Graph distillation for action detection with privileged modalities. In: ECCV (2018)
39. Ma, W.C., Huang, D.A., Lee, N., Kitani, K.M.: Forecasting interactive dynamics of pedestrians with fictitious play. In: CVPR (2017)
40. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
41. Makansi, O., Ilg, E., Cicek, O., Brox, T.: Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In: CVPR (2019)

42. Mangalam, K., Adeli, E., Lee, K.H., Gaidon, A., Niebles, J.C.: Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision. arXiv preprint arXiv:1911.01138 (2019)
43. Manh, H., Alaghband, G.: Scene-lstm: A model for human trajectory prediction. arXiv preprint arXiv:1808.04018 (2018)
44. Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.C., Lee, J.T., Mukherjee, S., Aggarwal, J., Lee, H., Davis, L., et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: CVPR (2011)
45. Pellegrini, S., Ess, A., Van Gool, L.: Improving data association by joint modeling of pedestrian trajectories and groupings. In: ECCV (2012)
46. Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T.S., Wang, Y.: Unrealcv: Virtual worlds for computer vision. In: ACM Multimedia (2017)
47. Rhinehart, N., Kitani, K.M.: First-person activity forecasting with online inverse reinforcement learning. In: ICCV (2017)
48. Rhinehart, N., Kitani, K.M., Vernaza, P.: R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In: ECCV (2018)
49. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: ECCV (2016)
50. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: ECCV (2016)
51. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
52. Ruiz, N., Schulter, S., Chandraker, M.: Learning to simulate. arXiv preprint arXiv:1810.02513 (2018)
53. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. arXiv preprint arXiv:1806.01482 (2018)
54. Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., Savarese, S.: Car-net: Clairvoyant attentive recurrent network. In: ECCV (2018)
55. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics. pp. 621–635. Springer (2018)
56. de Souza, C.R., Gaidon, A., Cabon, Y., López, A.M.: Procedural generation of videos to train deep action recognition networks. In: CVPR. pp. 2594–2604. IEEE (2017)
57. Styles, O., Ross, A., Sanchez, V.: Forecasting pedestrian trajectory with machine-annotated training data. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 716–721. IEEE (2019)
58. Styles, O., Guha, T., Sanchez, V.: Multiple object forecasting: Predicting future object locations in diverse environments. arXiv preprint arXiv:1909.11944 (2019)
59. Sun, C., Karlsson, P., Wu, J., Tenenbaum, J.B., Murphy, K.: Stochastic prediction of multi-agent interactions from partial observations. arXiv preprint arXiv:1902.09641 (2019)
60. Sun, S.H., Huh, M., Liao, Y.H., Zhang, N., Lim, J.J.: Multi-view to novel view: Synthesizing novel views with self-learned confidence. In: ECCV (2018)
61. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
62. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)

63. Vapnik, V., Izmailov, R.: Learning using privileged information: similarity control and knowledge transfer. Journal of machine learning research **16**(2023-2049), 2 (2015)
64. Varol, G., Laptev, I., Schmid, C., Zisserman, A.: Synthetic humans for action recognition from unseen viewpoints. arXiv preprint arXiv:1912.04070 (2019)
65. Wang, Y., Jiang, L., Yang, M.H., Li, L.J., Long, M., Fei-Fei, L.: Eidetic 3d lstm: A model for video prediction and beyond. In: ICLR (2019)
66. Wu, Y., Jiang, L., Yang, Y.: Revisiting embodiedqa: A simple baseline and beyond. arXiv preprint arXiv:1904.04166 (2019)
67. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR (2019)
68. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NeurIPS (2015)
69. Xue, H., Huynh, D.Q., Reynolds, M.: Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In: WACV (2018)
70. Yagi, T., Mangalam, K., Yonetani, R., Sato, Y.: Future person localization in first-person videos. In: CVPR (2018)
71. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687 (2018)
72. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
73. Zeng, X., Liu, C., Wang, Y.S., Qiu, W., Xie, L., Tai, Y.W., Tang, C.K., Yuille, A.L.: Adversarial attacks beyond the image space. In: CVPR (2019)
74. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
75. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In: CVPR (2019)
76. Zhang, Y., Wei, X., Qiu, W., Xiao, Z., Hager, G.D., Yuille, A.: Rsa: Randomized simulation as augmentation for robust human action recognition. arXiv preprint arXiv:1912.01180 (2019)
77. Zhang, Y., Gibson, G.M., Hay, R., Bowman, R.W., Padgett, M.J., Edgar, M.P.: A fast 3d reconstruction system with a low-cost camera accessory. Scientific reports **5**, 10909 (2015)
78. Zhao, T., Xu, Y., Monfort, M., Choi, W., Baker, C., Zhao, Y., Wang, Y., Wu, Y.N.: Multi-agent tensor fusion for contextual trajectory prediction. In: CVPR (2019)
79. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR (2017)
80. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: ICRA. pp. 3357–3364. IEEE (2017)