SimAug: Learning Robust Representations from Simulation for Trajectory Prediction

Junwei Liang[1][0000−0003−2219−5569], Lu Jiang[2], and Alexander Hauptmann[1]

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

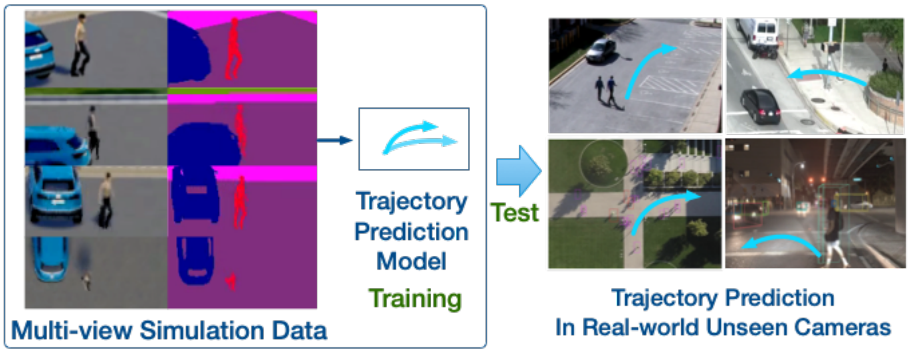
Multi-view Simulation Data
Trajectory Prediction Model
Training
Test
Trajectory Prediction In Real-world Unseen Cameras

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 studies the problem of predicting future trajectories of people in unseen cameras of novel scenarios and views. We approach this problem through the real-data-free setting in which the model is trained only on 3D simulation data and applied out-of-the-box to a wide variety of real cameras. We propose a novel approach to learn robust representation through augmenting the simulation training data such that the representation can better generalize to unseen real-world test data. The key idea is to mix the feature of the hardest camera view with the adversarial feature of the original view. We refer to our method as *SimAug*. We show that *SimAug* achieves promising results on three real-world benchmarks using zero real training data, and state-of-the-art performance in the Stanford Drone and the VIRAT/ActEV dataset when using in-domain training data. Code and models are released at https://next.cs.cmu.edu/simaug.

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

1 Introduction

Future trajectory prediction [27,1,19,38,56,31,37] is a fundamental problem in video analytics, which aims at forecasting a pedestrian's future path in the video in the next few seconds. Recent advancements in future trajectory prediction have been successful in a variety of vision applications such as self-driving vehicles [4,7,8], safety monitoring [38], robotic planning [50,51], among others.

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 [47,35,1,19,32,40], drone cameras [56,13,33], ground-level egocentric cameras [73,50,61], or dash cameras [45,60,8]. 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 across 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, resulting in a procedure that is not only expensive but also tardy in deploying the model.

An ideal model should be able to disentangle human behavioral dynamics from specific camera views, positions, and scenes. It should produce robust trajectory prediction despite the variances in these camera settings. Motivated by this idea, in this work, we learn a robust representation for future trajectory prediction that can generalize to unseen video cameras. Different from existing works, we study a *real-data-free* setting where a model is trained only on synthetic data but tested, out of the box, on unseen real-world videos, without further re-training or fine-tuning the model. Following the success of learning from simulation [55,59,67,79,15,52], our synthetic data is generalized from a 3D simulator, called CARLA [14], which anchors to the static scene and dynamic elements in the VIRAT/ActEV videos [47]. By virtue of the 3D simulator, we can generate multiple views and pixel-precise semantic segmentation labels for each training trajectory, as illustrated in Figure 1. Meanwhile, following the previous works [56,37], scene semantic segmentation is used instead of RGB pixels to alleviate the influence of different lighting conditions, scene textures, subtle noises produced by camera sensors, etc. At test time, we extract scene features from real videos using pretrained segmentation models. The use of segmentation features is helpful but is insufficient for learning robust representation for future trajectory prediction.

To tackle this issue, we propose a novel data augmentation method called *SimAug* to augment the features of the simulation data with the goal of learning robust representation to various semantic scenes and camera views in real videos. To be specific, first, after representing each training trajectory by high-level scene semantic segmentation features, we 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 overcoming the "hardest" view to which the model has learned. Following [24,22], 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 backbone model is built on a recent work called Multiverse [37]. The final model is trained to minimize the empirical vicinal risk over the distribution of augmented trajectories. Our method is partially inspired by recent robust deep learning methods using adversarial training [29,12], Mixup [77], and MentorMix [22].

We empirically validate our model, which is trained only on simulation data, on three real-world benchmarks for future trajectory prediction: VIRAT/ActEV [47,2], Stanford Drone [53], and Argoverse [8]. These benchmarks represent three distinct 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 data augmentation, adversarial learning, and imitation learning. Notably, our method achieves better results compared to the state-of-the-art on the VIRAT/ActEV and Stanford Drone benchmark. Our code and models are released at `https://next.cs.cmu.edu/simaug`. To summarize, our contribution is threefold:

- We study a new setting of future trajectory prediction in which the model is trained only on synthetic data and tested, out of the box, on any unseen real video with novel views or scenes.
- We propose a novel 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 establishes new state-of-the-art results on three public benchmarks.

2 Related Work

Trajectory prediction. Recently there is a large body of work on predicting person future trajectories in a variety of scenarios. Many works [1,72,78,38,37,56] focused on modeling person motions in videos recorded with stationary cameras. Datasets like VIRAT/ActEV [47], ETH/UCY [32,40] and Stanford Drone [53] have been used for evaluating pedestrian trajectory prediction. For example, Social-LSTM [1] added social pooling to model nearby pedestrian trajectory patterns. Social-GAN [19] added an 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. [27] used Inverse Reinforcement Learning to forecast human trajectory. SoPhie [56] combined deep neural network features from scene semantic segmentation model and generative adversarial network (GAN) using attention to model person trajectory. More recent works [28,73,42,38] have attempted to predict person paths by utilizing individuals' visual features instead of considering them as points in the scene. For example, Liang et al. [37]

proposed to use abstract scene semantic segmentation features for better generalization. Meanwhile, many works [31,57,4,21,81,44,33,51] have been proposed for top-down view videos for trajectory prediction. Notably, the Stanford Drone Dataset (SDD) [53] is used in many works [56,13,33] for trajectory prediction with drone videos. Other works have also looked into pedestrian prediction in dashcam videos [45,60,28,31] and first-person videos [73,61]. Many vehicle trajectory datasets [6,8,74] 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 [80,34,58,14,52,54,20], many research works have used 3D simulation for training and evaluating real-world tasks [15,59,69,83,62,37,63,3,26,9]. Many works [48,15,59] were proposed to use data generated from 3D simulation for video object detection, tracking, and action recognition analysis. For example, Sun et al. [62] proposed a forecasting model by using a gaming simulator. AirSim [58] and CARLA [14] were proposed for robotic autonomous controls for drones and vehicles. Zeng et al. [76] proposed to use 3D simulation for adversarial attacks. RSA [79] used randomized simulation data for human action recognition. The ForkingPaths dataset [37] 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.

Robust Deep Learning. Traditional domain adaptation approaches [5,16,65,25] may not be applicable as our target domain is considered "unseen" during training. Methods for learning using privileged information [30,66,39,41] is not applicable for a similar reason. Closest to ours is robust deep learning methods. In particular, our approach is inspired by the following methods: (i) *adversarial training* [18,43,70,76] to defend the adversarial attacks generated on-the-fly during training using gradient-based methods [43,18,64,11]; (ii) data augmentation methods to overcome unknown variances between training and test examples such as Mixup [77], MentorMix [22], AugMix [12], etc; (iii) example re-weighting or selection [24,49,23,36,46] to mitigate network memorization. Different from prior work, ours uses 3D simulation data as a new perspective for data augmentation and is carefully designed for future trajectory prediction.

3 Approach

In this section, we describe our approach to learn robust representation 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 the 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 historical video frames $V_{1:h}$ of the past h steps and the past agent locations $L_{1:h}$ in training, we learn a probabilistic model on

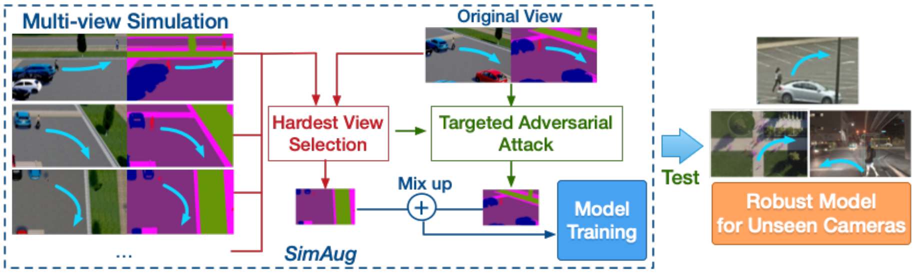

Fig. 2: Overview of our method *SimAug* that is trained on simulation and tested on real unseen videos. Each training trajectory is represented by multi-view segmentation features extracted from the simulator. *SimAug* mixes the feature of the hardest camera view with the adversarial feature of the original view.

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 $L_{h+1:T} = \{y_{h+1}, \dots, y_T\}$, where y_t is the location coordinates. As the test real 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 [37] that are semi-manually recreated from the VIRAT/ActEV benchmark that projects 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 tasks as suggested in previous works [15,54,37,79].

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 $\mathcal{S} = \{(V^{(i)}_{1:T}, L^{(i)}_{1:T})\}_{i=1}^{|\mathcal{S}|}$ denote a set of additional views for the same trajectory. In our experiments, we use four camera parameters pre-specified in [37], including three 45-degree views and one top-down view. We use a total of 4 scenes shown in Fig. 3. The ground-truth location varies under different camera views i.e. $L^{(i)}_{1:T} \neq L^{(j)}_{1:T}$ for $i \neq j$. Note that these camera positions and angles are defined in [37] specifically for VIRAT/ActEV dataset. The top-down view cameras in Stanford Drone dataset [53] are still considered unseen to the model since the scenes and camera positions are quite different.

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

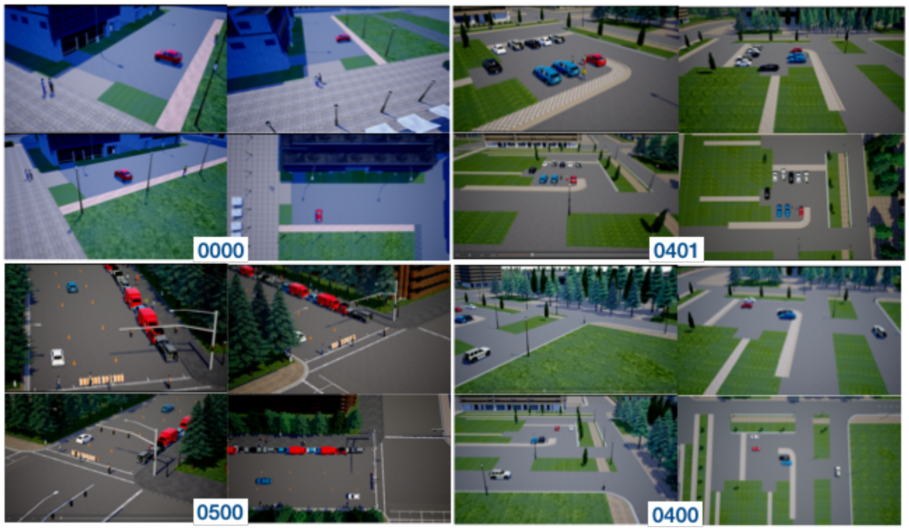

Fig. 3: Visualization of the multi-view 3D simulation data used in *SimAug* training. Data generation process is described in Section 3.2. We use 4 camera views from 4 scenes defined in [37]. "0400" and "0401" scene have overlapping views. The top-left views are the original views from VIRAT/ActEV dataset.

time, we extract the semantic segmentation feature from real videos using a pre-trained model with the same number of class labels per pixel. To be specific, we use the Deeplab model [10] trained on the ADE20k [82] 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, following previous works [38,37,13,56]. As we show in our experiments, the use of segmentation features is helpful but is still insufficient for learning the robust representation.

3.3 Multi-view Simulation Augmentation (*SimAug*)

In this subsection, we first describe *SimAug* for learning robust representations. Our trajectory prediction backbone model is built on the Multiverse model [37] and will be discussed in Section 3.4.

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 the scene semantic feature of view i at time t. $L_{1:T}^{(i)}$ is a sequence of ground-truth locations for the i-th view.

Each time given a camera view, we use it as an anchor to search for the "hardest" view that is most inconsistent with what the model has learned. Inspired by [24], we use the classification loss as the criteria and compute:

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

where δ is the ℓ_∞-bounded random perturbation applied to the input features. $\mathcal{L}_{\mathrm{cls}}$ is the location classification loss used in our backbone Multiverse model and will be discussed in the next subsection.

Then for the original view, we generate an adversarial trajectory by the targeted-FGSM attack [29]:

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

where ϵ is the hyper-parameter. The attack tries to make the model predict the future locations in the selected "hardest" camera view rather than the original view. In essence, the resulting adversarial feature is "warped" to the "hardest" camera view by a small perturbation. By defending against such adversarial trajectory, our model learns representations that are robust against variances in camera views.

Finally, we mix up the trajectory locations of the selected view and the adversarial trajectory locations by a convex combination function [77] over their features and one-hot location labels.

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

where $[y_{h+1}, \cdots, y_T] = L_{h+1:T}$ are the ground-truth locations of the original view. The one-hot$(\cdot)$ function projects the location in xy coordinates into an one-hot embedding over the predefined grid used in our backbone trajectory prediction model. Please find the details in [37]. Following [77], λ is drawn from a Beta distribution Beta(α, α) controlled by the hyper-parameter α.

The 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 trajectories to overcome (i) feature perturbations (Step 3 and 5) (ii) targeted adversarial attack (Step 5), and (iii) the "hardest" feature from other views (Step 4). By the mix-up operation in Eq. (3), our model is trained to minimize the empirical vicinal risk over a new distribution constituted by the generated augmented trajectories, which is proved to be useful in improving model robustness to real-world distributions under various settings [22].

3.4 Backbone Model for Trajectory Prediction

We employ Multiverse [37] as our backbone network, a state-of-the-art multi-future trajectory prediction model. Although we showcase the use of *SimAug* to

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 $\mathcal{S} = \{(V_{1:T}^{(i)}, L_{1:T}^{(i)})\}$;
3      Compute the loss for each camera view $\mathcal{L}_{\text{cls}}(V_{1:h} + \delta, L_{h+1:T}^{(j)})$;
4      Select the view with the largest loss $j^*$ by Eq. (1) ;
5      Generate an adversarial trajectory $V_{1:h}^{adv}$ by Eq. (2);
6      Mix up $(V_{1:h}^{adv}, L_{h+1:T})$ and $(V_{1:h}^{(j^*)}, L_{h+1:T}^{(j*)})$ by Eq. (3);
7      Compute $\mathcal{L}_{\text{cls}}$ over the augmented trajectory $(V_{1:h}^{aug}, L_{h+1:T}^{aug})$ from Step 6;
8  end
9  return averaged $\mathcal{L}_{\text{cls}}$ over the augmented trajectories
```

improve the robustness of Multiverse, *SimAug* is a general approach that can be applied to other trajectory prediction 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 as semantic segmentation feature of size $H \times W \times K$ where $K = 13$ is the total number of class labels as in [37,38]. As discussed in the previous section, we use *SimAug* to generate augmented trajectories $(V_{1:h}^{aug}, L_{1:h}^{aug})$ as our training features.

History Encoder. A convolutional RNN [71,68] 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 a concatenation of the last hidden state and the historical 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:

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

where f_c is the convolutional recurrent neural network (ConvRNN) with graph attention proposed in [37] and H_t^c 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:

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

where f_o is a separate ConvRNN and H_t^o 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 location 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 and scenes: the VIRAT/ActEV [2,47] dataset, the Stanford Drone Dataset (SDD) [53], 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

Following prior works [1,37], we utilize two common metrics for trajectory prediction evaluation. 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* (minADE_K): for each true trajectory sample i, we select the closest K predictions, and then measure its average error:

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

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:

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

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. This is an intermediate metric and hence is less indicative than the $\mathrm{minADE_K}$ and $\mathrm{minFDE_K}$.

4.2 Main Results

Dataset & Setups. We compare *SimAug* with classical data augmentation methods as well as 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 the same benchmarks. Real videos are not allowed to be used during training except in our finetuning experiments. For VIRAT/ActEV and SDD, we use the standard test split as in [38,37] and [56,13], respectively. For Argoverse, we use the official validation set from the 3D tracking task, and the videos from the "ring_front_center" camera are used.

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

Following the setting in previous works [38,1,19,1,19,56,44,37,13], the models observe 3.2 seconds (8 frames) of every pedestrian and predict the future 4.8 seconds (12 frames) of the person trajectory. We use the pixel values for the trajectory coordinates as it is done in [73,38,31,7,33,44,4,21,13]. By default, we evaluate the top $K = 1$ future trajectory prediction of all models.

Baseline methods. We compare *SimAug* with the following baseline methods for learning robust representations. All methods are built on the same backbone network and trained using the same simulation training data. *Base Model* is the trajectory prediction model proposed in [37]. *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]. Random labels are used for the targeted-FGSM attack. *Projected Gradient Descent (PGD)* is learned with an iterative adversarial learning method [43,70]. The number of iterations is set to 10 and other hyper-parameters following [70].

Implementation Details. We use $\alpha = 0.2$ for the Beta distribution in Eq (3) and we use $\epsilon = \delta = 0.1$ in Eq (2). As the random perturbation is small, we do not normalize the perturbed features and the normalized features yield comparable results. All models are trained using Adadelta optimizer [75] 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 [37].

Quantitative Results. Table 1 shows the evaluation results. Our method performs favorably against other baseline methods across all evaluation metrics on 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 stemming from the effective use of multi-view simulation 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 future trajectory prediction in unseen cameras. Note as the baseline methods use the same features as ours, the results indicate the use of segmentation features is insufficient for learning robust representations.

Qualitative Analysis. We visualize outputs of the base model with and without *SimAug* in Fig. 4. We show visualizations on all three datasets. In each image, the yellow trajectories denote historical 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 for turnings (Fig. 4 1a., 3a.) while without it the model sometimes predicts the wrong turns (Fig. 4 1b., 1c., 2a., 3a., 3b.). In addition, the length of SimAug predictions is more accurate (Fig. 4 1d., 2b., 2c., 2d.).

Table 1: Comparison to the standard data augmentation method and recent adversarial learning methods on three datasets. We report three metrics: Grid_Acc(↑)/minADE$_1$(↓)/minFDE$_1$(↓). The units of ADE/FDE are pixels. All methods are built on the same backbone model in [37] and trained using the same multi-view simulation data described in Section 3.2.

Method	VIRAT/ActEV	Stanford Drone	Argoverse
Base Model [37]	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 [43,70]	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 [1], Social-GAN [19], DESIRE [31], and SoPhie [56]. We also compare with the imitation learning model, IDL [33], and the inverse reinforcement learning model, P2T$_{\text{IRL}}$ [13] for trajectory prediction on the Stanford Drone Dataset. Following previous works, we evaluate the minimal errors over $K = 20$ predictions.

Results & Analysis. The results are shown in Table 2 (a), where *SimAug* is built on top of the *Multiverse* model. As it shows, *SimAug* model trained only on the 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

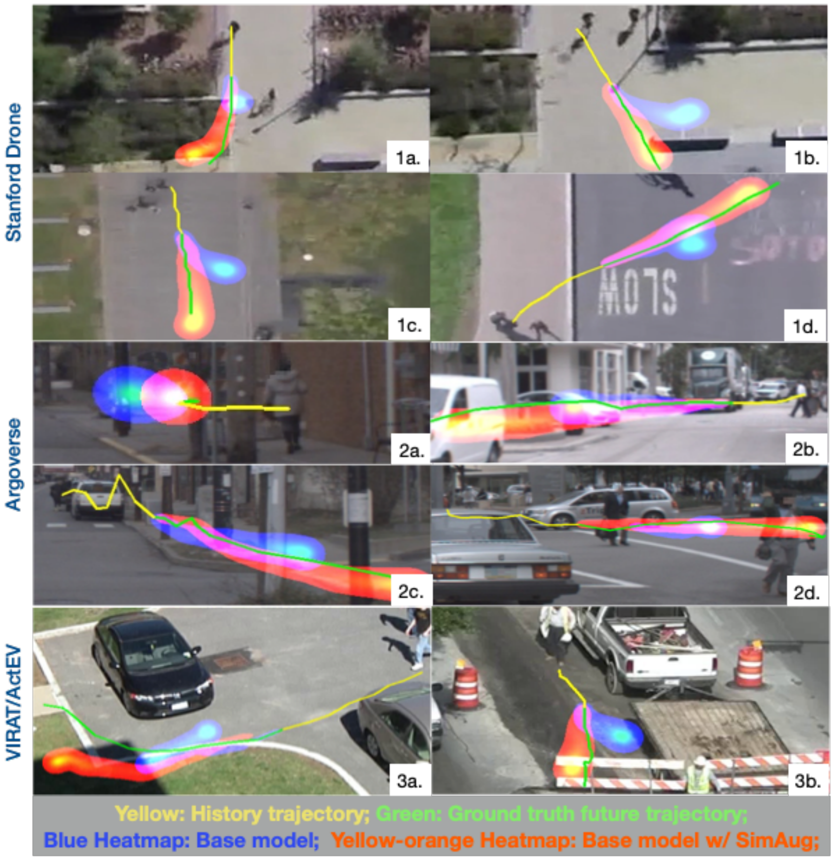
Stanford Drone
Argoverse
VIRAT/ActEV
1a.
1b.
1c.
1d.
2a.
2b.
2c.
2d.
3a.
3b.
Yellow: History trajectory; Green: Ground truth future trajectory;
Blue Heatmap: Base model; Yellow-orange Heatmap: Base model w/ SimAug;

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

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 future trajectory prediction in unseen cameras.

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 [37], we compute the errors for the top $K = 1$ prediction. Experimental results are shown in Table 2 (b), where all models in the top four rows are trained on the real-world training videos in VIRAT/ActEV. Our model trained on simulation data achieves competitive performance and outperforms *Multiverse* [37] model that is trained on the same

Table 2: State-of-the-art comparison on the Stanford Drone Dataset (SDD) and on the VIRAT/ActEV dataset. Numbers are minimal errors over 20 predictions for SDD and minimal errors over the top prediction for VIRAT/ActEV. Baseline models are all trained on real videos and their numbers are taken from [56,13]. "SimAug" is trained only using simulation data and "SimAug*" is further fine-tuned on the real training videos. "Multiverse" in the bottom rows is trained only with simulation data.

Method	minADE$_{20}$(↓)	minFDE$_{20}$ (↓)
Social-LSTM [1]	31.19	56.97
Social-GAN [19]	27.25	41.44
DESIRE [31]	19.25	34.05
SoPhie [56]	16.27	29.38
Multiverse [37]	14.78	27.09
IDL [33]	13.93	24.40
P2T$_{\text{IRL}}$ [13]	12.58	22.07
SimAug	12.03	23.98
SimAug*	**10.27**	**19.71**

(a) Stanford Drone Dataset

Method	minADE$_1$(↓)	minFDE$_1$ (↓)
Social-LSTM [1]	23.10	44.27
Social-GAN [19]	30.42	60.70
Next [38]	19.78	42.43
Multiverse [37]	18.51	35.84
Multiverse [37]	22.94	43.35
SimAug	21.73	42.22
SimAug*	**17.96**	**34.68**

(b) VIRAT/ActEV

data. With fine-tuning, which means using exactly the same training data without any extra annotation of real trajectories compared to [1,19,38,37], 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 3, where the top-1 prediction is used in the evaluation. We verify four key design choices by removing each, at a time, from the full model. The results show that by introducing viewpoint selection (Eq. (1)) and adversarial perturbation (Eq. (2)), our method improves model generalization.

(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 the top-down view because it is similar to the ones in the SDD dataset. As we see, the performance drops due to the fewer number of data and less diverse views, suggesting that we should use more views in augmentation (which is effortless to do in 3D simulators).

(2) *Random perturbation:* We test our model without random perturbation on the original view trajectory samples by setting $\delta = 0$ in Eq. (1). This leads to the performance drop 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 $V^{adv}_{1:h} = V_{1:h}$. This is similar to applying the Mixup

method [77] to two views in the feature space. The performance drops 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, especially on the Stanford Drone dataset, verifying the effectiveness of this design.

Table 3: Performance on ablated versions of our method on three benchmarks. We report the minADE$_1$(↓)/minFDE$_1$(↓) metrics.

Method	VIRAT/ActEV	Stanford Drone	Argoverse
SimAug full model	21.7 / 42.2	15.7 / 30.2	67.9 / 175.6
- top-down view data	22.8 / 43.6	18.4 / 35.6	68.4 / 178.3
- random perturbation	23.6 / 43.8	18.7 / 35.6	69.1 / 180.2
- adversarial attack	23.1 / 43.8	17.4 / 32.9	68.0 / 177.5
- view selection	23.0 / 42.9	19.6 / 38.2	68.6 / 177.0

5 Conclusion

In this paper, we have introduced *SimAug*, a novel simulation data augmentation method to learn robust representations for trajectory prediction. Our model is trained only on 3D simulation data and applied out-of-the-box to a wide variety of real video cameras with novel views or scenes. 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 learning robust representation for trajectory prediction with limited or zero training data. Other directions to deal with camera view dependence include using a homography matrix, which may require an additional step of manual or automatic calibration of multiple cameras. We leave them for future work.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments, and Google Cloud for providing GCP research credits. This research was supported by NSF grant IIS-1650994, the financial assistance award 60NANB17D156 from NIST, and the 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, or the U.S. Government.

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, 3, 9, 10, 11, 13
2. Awad, G., Butt, A., Curtis, K., Fiscus, J., Godil, A., Smeaton, A.F., Graham, Y., Kraaij, W., Quénot, 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, 9
3. Bak, S., Carr, P., Lalonde, J.F.: Domain adaptation through synthesis for unsupervised person re-identification. In: ECCV (2018) 4
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) 2, 4, 10
5. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR (2017) 4
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) 4
7. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449 (2019) 2, 10
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) 2, 3, 4, 9
9. Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., Tian, Q.: Data-free learning of student networks. In: ICCV (2019) 4
10. 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) 6
11. Cheng, Y., Jiang, L., Macherey, W.: Robust neural machine translation with doubly adversarial inputs. ACL (2019) 4
12. Cheng, Y., Jiang, L., Macherey, W., Eisenstein, J.: Advaug: Robust data augmentation for neural machine translation. In: ACL (2020) 3, 4
13. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735 (2020) 2, 4, 6, 10, 11, 13
14. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938 (2017) 2, 4, 5
15. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: CVPR (2016) 2, 4, 5
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) 4
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) 3
18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014) 2, 4, 10, 11

19. Gupta, A., Johnson, J., Savarese, Li Fei-Fei, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: CVPR (2018) 2, 3, 10, 11, 13
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) 4
21. Hong, J., Sapp, B., Philbin, J.: Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: CVPR (2019) 4, 10
22. Jiang, L., Huang, D., Liu, M., Yang, W.: Beyond synthetic noise: Deep learning on controlled noisy labels. In: ICML (2020) 3, 4, 7
23. Jiang, L., Meng, D., Zhao, Q., Shan, S., Hauptmann, A.G.: Self-paced curriculum learning. In: AAAI (2015) 4
24. 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. ICML (2018) 3, 4, 7
25. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR (2019) 4
26. Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S.: Meta-sim: Learning to generate synthetic datasets. In: ICCV (2019) 4
27. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: ECCV (2012) 2, 3
28. Kooij, J.F.P., Schneider, N., Flohr, F., Gavrila, D.M.: Context-based pedestrian path prediction. In: ECCV (2014) 3, 4
29. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. ICLR (2017) 3, 7
30. Lambert, J., Sener, O., Savarese, S.: Deep learning under privileged information using heteroscedastic dropout. In: CVPR (2018) 4
31. 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) 2, 4, 10, 11, 13
32. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum. pp. 655–664. Wiley Online Library (2007) 2, 3
33. Li, Y.: Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In: CVPR (2019) 2, 4, 10, 11, 13
34. 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) 4
35. 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) 2
36. Liang, J., Jiang, L., Meng, D., Hauptmann, A.G.: Learning to detect concepts from webly-labeled video data. In: IJCAI (2016) 4
37. Liang, J., Jiang, L., Murphy, K., Yu, T., Hauptmann, A.: The garden of forking paths: Towards multi-future trajectory prediction. In: CVPR (2020) 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
38. 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) 2, 3, 6, 8, 10, 13
39. Lopez-Paz, D., Bottou, L., Schölkopf, B., Vapnik, V.: Unifying distillation and privileged information. arXiv preprint arXiv:1511.03643 (2015) 4

40. Luber, M., Stork, J.A., Tipaldi, G.D., Arras, K.O.: People tracking with human motion predictions from social forces. In: ICRA (2010) 2, 3
41. Luo, Z., Hsieh, J.T., Jiang, L., Carlos Niebles, J., Fei-Fei, L.: Graph distillation for action detection with privileged modalities. In: ECCV (2018) 4
42. Ma, W.C., Huang, D.A., Lee, N., Kitani, K.M.: Forecasting interactive dynamics of pedestrians with fictitious play. In: CVPR (2017) 3
43. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017) 4, 10, 11
44. 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) 4, 10
45. 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) 2, 4
46. Northcutt, C.G., Jiang, L., Chuang, I.L.: Confident learning: Estimating uncertainty in dataset labels. arXiv preprint arXiv:1911.00068 (2019) 4
47. 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) 2, 3, 9
48. 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) 4
49. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: ICML (2018) 4
50. Rhinehart, N., Kitani, K.M.: First-person activity forecasting with online inverse reinforcement learning. In: ICCV (2017) 2
51. Rhinehart, N., Kitani, K.M., Vernaza, P.: R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In: ECCV (2018) 2, 4
52. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: ECCV (2016) 2, 4
53. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: ECCV (2016) 3, 4, 5, 9
54. 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) 4, 5
55. Ruiz, N., Schulter, S., Chandraker, M.: Learning to simulate. arXiv preprint arXiv:1810.02513 (2018) 2
56. 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) 2, 3, 4, 6, 10, 11, 13
57. Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., Savarese, S.: Car-net: Clairvoyant attentive recurrent network. In: ECCV (2018) 4
58. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics (2018) 4
59. Souza, C.R., Gaidon, A., Cabon, Y., López, A.M.: Procedural generation of videos to train deep action recognition networks. In: CVPR (2017) 2, 4
60. 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) 2, 4

61. Styles, O., Guha, T., Sanchez, V.: Multiple object forecasting: Predicting future object locations in diverse environments. arXiv preprint arXiv:1909.11944 (2019) 2, 4
62. 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) 4
63. 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) 4
64. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017) 4
65. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017) 4
66. Vapnik, V., Izmailov, R.: Learning using privileged information: similarity control and knowledge transfer. Journal of machine learning research **16**(2023-2049), 2 (2015) 4
67. Varol, G., Laptev, I., Schmid, C., Zisserman, A.: Synthetic humans for action recognition from unseen viewpoints. arXiv preprint arXiv:1912.04070 (2019) 2
68. 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) 8
69. Wu, Y., Jiang, L., Yang, Y.: Revisiting embodiedqa: A simple baseline and beyond. IEEE Transactions on Image Processing **29**, 3984–3992 (2020) 4
70. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR (2019) 4, 10, 11
71. 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) 8
72. Xue, H., Huynh, D.Q., Reynolds, M.: Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In: WACV (2018) 3
73. Yagi, T., Mangalam, K., Yonetani, R., Sato, Y.: Future person localization in first-person videos. In: CVPR (2018) 2, 3, 4, 10
74. 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) 4
75. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012) 10
76. 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) 4
77. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018) 3, 4, 7, 14
78. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In: CVPR (2019) 3
79. 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) 2, 4, 5
80. 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) 4
81. 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) 4

82. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR (2017) 6
83. 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 (2017) 4