T2FPV: Constructing High-Fidelity First-Person View Datasets From Real-World Pedestrian Trajectories

Benjamin Stoler\textsuperscript{1}, Meghdeep Jana\textsuperscript{2}, Soonmin Hwang\textsuperscript{3}, and Jean Oh\textsuperscript{3}

Abstract—Predicting pedestrian motion is essential for developing socially-aware robots that interact in a crowded environment. While the natural visual perspective for a social interaction setting is an egocentric view, the majority of existing work in trajectory prediction has been investigated purely in the top-down trajectory space. To support first-person view trajectory prediction research, we present T2FPV, a method for constructing high-fidelity first-person view datasets given a real-world, top-down trajectory dataset; we showcase our approach on the ETH/UCY pedestrian dataset to generate the egocentric visual data of all interacting pedestrians. We report that the bird’s-eye view assumption used in the original ETH/UCY dataset, i.e., an agent can observe everyone in the scene with perfect information, does not hold in the first-person views; only a fraction of agents are fully visible during each 20-timestep scene used commonly in existing work. We evaluate existing trajectory prediction approaches under varying levels of realistic perception—displacement errors suffer a 356% increase compared to the top-down, perfect information setting. To promote research in first-person view trajectory prediction, we release our T2FPV-ETH dataset and software tools.

I. INTRODUCTION

As more and more autonomous robots are anticipated to interact with people in shared environments, trajectory prediction in robotics including navigation among human crowds \cite{2, 3, 4, 5, 6} and unmanned aerial vehicles \cite{7} has become increasingly popular in the research community, as well as among various industry and military stakeholders. In particular, predicting pedestrian motion is essential for developing socially-aware robots that interact in a crowded environment. Existing state-of-the-art (SOTA) trajectory prediction algorithms leverage datasets such as the ETH/UCY pedestrian dataset that provide full trajectory information of all pedestrians in a bird’s-eye view (BEV) scene \cite{1}. However, bird’s-eye view is an unrealistic view for agents navigating in the real-world; agents generally rely on egocentric, first-person view (FPV) sensing for these tasks. A realistic setting also includes limited field-of-view (FOV), occlusions, and changes in perspective and orientation of the ego-agent.

Whereas it is relatively convenient to collect top-down data using an over-head camera, creating a first-person view counterpart is far more challenging due to several reasons. For instance, all participants in the scene would need to wear a camera sensor to record their egocentric views, as well as a location-recording sensor to establish their ground truth locations. Furthermore, such a setting is subject to psychological issues such as the observer (or Hawthorne) effect \cite{8}, where people’s behaviors in these experiments may not be entirely representative of a natural social interaction.

Existing real-world first-person view pedestrian datasets, such as \cite{4}, generally do not include other agents’ ground-truth world locations in the scenes. Other synthetic first-person view datasets \cite{5, 9} share a similar intuition as ours—however, their synthesized images’ quality is low fidelity, and they do not consider realistic perception. Our approach utilizes SEANavBench \cite{10}, a flexible high-fidelity simulation environment. In SEANavBench, to acquire first-
In this context, we propose Trajectory-to-First-Person Views (T2FPV), a method for constructing a first-person view version of data from a trajectory-only dataset by simulating the agents in high fidelity. Each agent follows their recorded trajectory with a simulated camera attached to them. To showcase our approach, we construct the T2FPV-ETH dataset based on the ETH/UCY trajectory dataset [1] as shown in Fig. [1].

We evaluate existing trajectory prediction approaches using a top-performing detection and tracking algorithm to show that the displacement errors on trajectory prediction increase by a large margin. As shown in Fig. [2] this is an increase of more than 1 meter (a 356% increase) compared to the top-down, perfect information setting.

Our main contributions are: 1) we propose a method for creating an egocentric view for each agent given a set of trajectories; 2) we generate the T2FPV-ETH dataset, a first-person view dataset that corresponds to the ETH/UCY dataset; 3) we perform experiments with increasingly realistic perception, and report the degraded performance of a top-performing trajectory prediction approach; 4) we show that layer normalization combined with social pooling generally improves trajectory prediction performance, especially in the first-person view setting; and 5) to promote research in first-person view trajectory prediction, we release our dataset and software tools.

II. RELATED WORK

Real-World First-Person Datasets. Various large-scale datasets provide video footage from an ego agent’s perspective. [11] is a dataset of egocentric video stream along with pose, acceleration and orientation information. [12] is a large-scale first-person view video dataset, with over 3500 hours of footage collected from various sources around the world. However, these datasets only provide the single perspective of an ego agent in each scenario, and lack the ground truth pose information and perspective of all other agents in the scene. Egocentric Basketball Motion Planning [13] provides a wearable camera perspective from multiple people in the scene. Nonetheless, these datasets are not focused on social navigation. They feature many instances of the ego agent walking by themselves or performing an unrelated task (such as carpentry, basketball, etc.) that inherently have different social contexts than navigating in public.

Synthetic Pedestrian Datasets and Simulation. Several recent works have generated synthetic data in simulations based on a corresponding real-world dataset. FvTraj [5] uses Unity to render FPV images from ground truth trajectory data [14], but these rendered images consist only of a flat ground plane, with no corresponding environment modeled. DeepSocNav [9] generates ego view depth images from ETH/UCY, with a low-fidelity environment model. However, they do not include the images from RGB cameras, which are far more common than depth sensors. Furthermore, DeepSocNav and FvTraj do not release any generated images or the in-house simulators. [15] and [10] are relatively high-fidelity simulation environments with scene constructions of ETH/UCY, built in Unreal Engine [16] and Unity [14] respectively, but both lack first-person views. Additionally, none of these synthetic datasets account for partial trajectories or realistic detection and tracking as a consequence of occlusions and limited FOV.

Trajectory Prediction. Recent work on trajectory prediction has mostly focused on top-down trajectory datasets such as ETH/UCY [1], SDD [17], and inD [18]. [2] uses LSTMs to jointly predict trajectories of all agents, incorporating pooled hidden-state information from neighbors as a social cue. Some approaches, such as AC-VRNN [19], use generative models within a VRNN [20], with social interactions incorporated via attentive hidden state refinement. [3] predicts socially plausible futures by training in an adversarial manner against a recurrent discriminator. Several works also leverage top-down images explicitly, whether in an RGB form or with added semantic segmentation [6], [21], [22]. SGNet [23] generates coarse step-wise goals to assist trajectory prediction in a sequential manner. [24] incorporates agent dynamics and environment information and forecasts using a graph-structured recurrent model. [4] utilizes FPV to model and predict the trajectory directly in pixel-space. [25] creates a spatial visual distribution of objects from FPV, and applies perception and ego-agent trajectory planning in a 2.5D coordinate system. Additionally, these FPV methods perform trajectory prediction on either a single target or only the ego agent themselves.

III. PROBLEM FORMULATION

A trajectory prediction problem using complete information is defined as follows: for $N$ pedestrians in a scene, we denote the position of each agent $i$ in the $xy$ ground-plane at time-step $t$ as $(x_i^t, y_i^t)$. Given the observed track histories $\{(x_i^1, y_i^1)|t = 1, 2, ..., T_{obs}\}$, the task is to predict the future paths $\{(x_i^1, y_i^1)|t = T_{obs}+1, T_{obs}+2, ..., T_{pred}\}$ for all agents $i$ in a given scene.

In this paper, we introduce a trajectory prediction task where each agent is to predict the trajectories of all agents in their view only using their egocentric information. Formally, let $\phi_i$ denote agent $i$'s FPV image at time step $t$ and $\Psi(i)$ denote the set of agents that are within agent $i$'s field of view. Then, agent $j \in \Psi(i)$ if agent $i$'s views $\{\phi_{i,t}|t = t', t'+1, ..., t'+k\}$ contain at least some $P$ pixels associated with agent $j$, for some $k$ time steps. For each agent $i$ in $1, ..., N$, the FPV trajectory prediction task is to predict the future trajectories of agent $i$ and all agents within agent $i$'s FOV, given FPV observations, $\{\phi_{i,t}|t = 1, ..., T_{obs}\}$, as well as their ego track history.
IV. TRAJECTORIES TO FIRST-PERSON VIEW

We describe how we construct first-person view data from a trajectory dataset, using the ETH/UCY dataset as an example.

A. Video and Annotation Generation

Our approach for creating FPV datasets from real-world trajectory datasets begins with generating videos and ground-truth annotations. We use the SEANavBench [10] simulation environment as a starting point for our simulation. SEANavBench consists of high-fidelity pre-modeled scenes for each location within ETH/UCY. We leave these scenes as unchanged as possible, for consistency with prior works using SEANavBench.

As in [5], we enforce a number of assumptions when rendering these tracks. For instance, we orient each pedestrian’s gaze with the direction they are traveling in, with spherical linear interpolation for smoother angle changes. Additionally, we mount a camera on each pedestrian at a fixed height of 1.6m from their base, and assign the following physical characteristics to the camera: 18mm focal length, 36×24mm sensor, and zero lens shift for the principal point. When rendered at our 640×480 resolution, this results in a vertical FOV of approximately 67°.

Using the above assumptions, we then render the first-person videos for every person following their track from the original dataset, as well as output an annotation for each agent at every frame. The videos consist of the RGB render, as well as an instance segmentation render, as shown in Fig. 1, where each object in the scene has been given a unique color. The annotations consist of the agent’s ID, pose information, and a list of what other agents can be seen in the camera’s view, i.e., the poses of all visible agents in both camera and world reference frame. This detection list is generated by utilizing the aforementioned segmentation mask to determine agent visibility.

B. Detection and Tracking

To assess the performance of SOTA trajectory prediction methods under a realistic setting, we employed an off-the-shelf object detector and tracker to produce the observations required for trajectory prediction. We used a 3D object detector [26] which is SOTA among recent image-only methods which do not require depth information [27], and a simple but effective probabilistic tracker [28]. We made changes to both approaches to produce reasonable detection and tracking results.

In DD3D [26], we set the parameters of feature map assignment to use thresholds that fit our ground truths appropriately. We also only used instances that are “visible” (as defined in Section IV-C.2), which helps to filter out heavily occluded instances. For the tracker [28], we changed the matching metric to use BEV IoU (Intersection-over-Union in top-down view) from Mahalanobis distance [29] to associate detections to tracks. We also applied the Kalman filter only to each instance’s 3D location and orientation and used state and observation noise covariances calculated from our ground truth data.

Following the common evaluation procedure as in the ETH/UCY trajectory prediction task, we trained one model for each of the five folds, using the other four folds as the training and validation sets respectively. We then produced tracking results on all ego videos from each fold’s test-set.

C. Data Loading

We explain several variations of data loading used in our dataset. Let us define a scene as a sequence of time steps where the agents in the environment have contiguous, full information provided over the durations of observation $T_{obs}$ and prediction $T_{pred}$, i.e., each 20-step scene in a sliding-window approach with a single step stride. Let a tracklet refer to the portion of the trajectory of an agent that is present in a scene.

We use the pre-processing as popularized in Social GAN [3], which consists of agent tracks in world coordinates taken at 2.5 FPS. For mediated perception approaches where vision-based approaches are used to pre-process FPV data into trajectories, we provide two versions of imperfect trajectories, one with synthetic noise and the other with vision-based detection and tracking.

Our dataset variations are defined as follows:

1) Bird’s-Eye View (BEV): Following the data loading process from Social GAN [3], BEV scenes are constructed as follows: in a potential scene, agents whose tracks start late, end early, or have other missing intermediate data points are thrown out. These scenes are then filtered to ensure that only scenes with at least two agents’ tracklets are included in the BEV set. When loading a batch, indexing is done at this scene-level granularity. In this set, agents’ visibility is ignored. Thus, two assumptions are used in BEV: 1) agents that are present partially during a scene are ignored, and 2) agents that are fully present during a scene are perfectly visible to every agent even if they are completely occluded to some agents.

2) First-Person View Ground Truth (FPV-GT): In transitioning from BEV to FPV, given a scene with $N$ agents, we now construct $N$ variations of the same scene, i.e., from each agent’s perspective.

To take realistic constraints and limitations such as occlusions and limited FOV, we redesign the scene as follows: First, we relax the full-duration assumption from BEV and include all of the partially observed agents that appear in at least $k$ of the first $T_{obs}$ time steps. Second, using the ground truth information as defined in Section IV-A, we only consider agents that are truly visible in the first-person view, with at least $P$ pixels visible. The number of tracklets in T2FPV-ETH, as seen in Table I, is based on $k = 3$ and $P = 100$. By addressing the two limitations of BEV, we establish FPV ground truth by emulating what a “perfect” first-person view detector and tracker would produce.

3) FPV-Noisy: To examine what slightly noisy detection and tracking would cause, we also provide a supplementary dataset by adding a small amount of synthetic noise
and other randomization to the ground truth visible agents, while leaving the ego agent track untouched. The sources of noise are defined as follows: 1) Each tracklet has a 1% chance of being dropped; 2) Each visible bounding box has a 10% chance of being dropped from the tracklet; 3) At each timestep in a detection sequence, there is a 2% chance of the detected agent being “lost”, i.e., assigned a new ID for subsequent frames; and 4) Each visible agent location in meters has Gaussian noise with \( \mu = 0 \) and \( \sigma = 0.05m \) applied. Note that the corresponding ground truth tracklets in the prediction phase \( t > T_{obs} \) have no such noise applied, even if the observation was fully dropped.

4) **FPV-Det**: This version of the dataset is processed identically to **FPV-GT**, except in the observation phase, the actual detection and tracking results from Section IV-B is substituted in. Note again, that the prediction ground truth is still unchanged from the original **FPV-GT** version.

D. Statistics

1) **BEV vs. FPV-GT**: Table I provides a high-level overview of the number of scenes and tracklets in each version of the T2FPV-ETH dataset, as created in Section IV-C. We note that this table demonstrates a data augmentation effect when using all perspectives in a BEV scene; a single ground-truth track is often observed by multiple other agents at once.

2) **FPV-Det**: We measure the detection and tracking performances of the SOTA methods we employed in Table II. For detection performance, we measure the standard average precision \( AP_{2D} \) in 2D image space and observe that it performs well. Also, we measure the localization quality of detected objects in 3D space by calculating IoU-based average precision in the top-down view \( AP_{BEV} \). Both metrics use the same IoU threshold of 0.5. The \( AP_{BEV} \) performance is worse than \( AP_{2D} \), which shows the challenge of image-based 3D detection. For tracking, we adopt two popular metrics from [30], Average Multi-Object Tracking Accuracy (AMOTA) and Precision (AMOTP). AMOTA combines false positives, missed targets, and identity switches, and AMOTP measures the misalignment between prediction and ground truth. Although “Univ” shows the worst performance because of the pedestrian density (Table I), the detector and tracker perform reasonably well in most cases, as shown qualitatively in Fig. 2.
B. Our Approach

To improve training stability, we employ layer normalization (LN) [31] after each hidden layer within the Multi-Layer Perceptron (MLP) blocks. This was motivated by works purporting its success in recurrent architectures, especially in recent Transformer-based approaches [32].

We also re-examined the methods of social awareness used, beyond GAT sub-networks [33]. This technique was especially important when trying to improve SGNet [23], since their approach has no method of social awareness built in. In particular, besides graph-based social methods, other works have attempted pooling approaches [2], [3]. We thus created a version of social pooling that incorporates lessons learned from GAT and hidden state-refinement approaches [33]: use a similarity adjacency matrix, based on closeness to other people in the scene at each timestep, to perform a weighted average pooling over others’ hidden states to refine each agent’s own hidden state.

VI. EXPERIMENTS

A. Evaluation Procedure

As in Social GAN, we evaluate trajectory predictions using a leave-one-out approach. For each of the five folds, models are trained and validated on data from four of them at a time. Then, the best model according to validation performance is tested on the entirety of the held-out fifth fold.

To examine how important perception quality is in trajectory prediction accuracy, we utilize a transfer setting, where models are trained using the ground truth scene groupings as in prior works (BEV), but then additionally tested on the three FPV settings (FPV-GT, FPV-Noisy, and FPV-Det), as described in Section IV-C.

B. Metrics

In the field of trajectory prediction, and especially for ETH/UCY, the most commonly used metrics are Average Displacement Error (ADE) and Final Displacement Error (FDE). These metrics can be easily computed on a per-agent basis, for ground truth future track \( \{ (x^t_i, y^t_i) | t = T_{obs+1}, T_{obs+2}, ..., T_{pred} \} \) and predicted future track \( \{ (\hat{x}^t_i, \hat{y}^t_i) | t = T_{obs+1}, T_{obs+2}, ..., T_{pred} \} \), for each agent \( i \) in the scene. The L2-distance at each time \( t \) is taken between \( (x^t_i, y^t_i) \) and \( (\hat{x}^t_i, \hat{y}^t_i) \): ADE is the average of these distances, while FDE is the final distance, at time \( T_{pred} \).

However, since we consider detection and tracking to be a core part of the challenge posed by this dataset, our metrics must extend beyond ADE/FDE and incorporate a sense of precision and recall regarding which agents should have been observed and predicted in the first place. This is made even more challenging by the fact that there is no guarantee of alignment of agent IDs between the observed tracks and ground truth tracks, even if the precision and recall is perfect.

To account for these issues, we use a variant of mean Average Precision [34] (mAP) for trajectory prediction, using ADE as a stand-in for both confidence and match quality.

Note that all approaches we consider utilize a “best-of-\( K \)” sample prediction strategy. This accounts for the fact that there are multiple socially valid predictions for agents in a scene, so feasible (but ultimately incorrect) predictions should be punished less. As a result, ADE and FDE are computed in a \( K \)-to-one manner, for each sample, and are
reduced to their minimum value before being passed to downstream mAP computation. We note that there are other metrics which could be utilized, including collision rate, social comfort level, path complexity, and many more [35]. We chose to focus on the core metrics of the tasks at hand, but suggest that future work in applying these metrics could provide helpful new insight.

VII. RESULTS

The ADE, FDE and mAP performance is shown for each fold in Table III and for each dataset variation in Fig. [---note that the x-axis is categorical, and points are connected only for improved visibility. In general, the FPV-GT version of the dataset proves to be a moderately hard setting, with an average ADE increase for the best performing SGNet variant of 14.8% for ADE. We hypothesize that this is mostly due to models having to predict partially observed detections, based on FOV and occlusion filtering. With the addition of a small amount of noise in FPV-Noisy, as well as actual FPV-Det detection and tracking results, the drop in performance is quite significant, i.e., 92.9% and 356% increase in ADE relative to BEV respectively on the top performing approach. All algorithms tested demonstrated similar behavior when evaluated on the different test-sets.

Regarding our proposed additions to existing approaches, applying LN seems to help near uniformly compared to each model’s non-LN counterpart. We suspect this could be due to the trajectory prediction methods being highly sensitive to hyper-parameter selection during training, and LN helps to “smooth” these dynamics [31]. Additionally, incorporating spatially-weighted average pooling (social pooling) seems to improve performance as well, in most cases more so than GAT sub-networks. One possible explanation for this is the added complexity of training GATs, due to their increase in resulting model size. Ultimately, we were able to noticeably improve upon our SOTA baseline selection of SGNet by incorporating these techniques. The improvement is most significant in FDE, indicating stronger long-term prediction capabilities. Note that mAP seems to be affected only marginally by algorithm variations, as they all share the same detection and tracking module. Approaches which utilize an end-to-end, rather than mediated, perception approach may exhibit more dramatic changes.

VIII. LIMITATIONS

Although SEANavBench is a high-fidelity environment, we do note that further effort in improving its realism could be useful. Realism could be enhanced not just by increasing the 3D-modeling asset and animation qualities, but also by further improving alignment between the reproduced scenery and the original locations. In particular, since we believe that performing Sim2Real transfer experiments on our tasks would be an interesting future direction, the resulting Sim2Real gap might be improved by increasing T2FPV quality.

Furthermore, we believe that significantly better results could be achieved by using models which leverage first-person view more explicitly, such as by incorporating image features, e.g. ResNet embeddings[36]. It could also be a fruitful direction to evaluate models which are trained directly on FPV-GT or FPV-Det, rather than our transfer setting.

IX. CONCLUSION

In existing work, pedestrian trajectory prediction has been mainly studied under a complete information assumption. In this paper, we introduce a first-person view trajectory prediction problem where agents would need to make predictions based on partial, imprecise information. To promote this research direction, we present T2FPV, a method for generating high-fidelity egocentric views for pedestrian navigation by leveraging existing real-world trajectory datasets. Our experiments show that top-performing baselines suffer a substantial decrease in performance in our proposed first-person view setting using a SOTA detection and tracking approach—ADE relative to performance when tested in the complete information, bird’s-eye view setting increases by 356%. We also report that using layer normalization and social pooling generally improves the prediction performance, especially in the first-person view setting. Our constructed T2FPV-ETH dataset provides a benchmark for trajectory prediction and pedestrian detection and tracking tasks in a more natural and realistic setting. We argue that this is an important direction to move toward in enabling robots to
navigated in the real world. To promote further research in first-person view trajectory prediction, we release our dataset and software tools to the public.

REFERENCES

[1] A. Ess, B. Leibe, K. Schindler, and L. van Gool, “A mobile vision system for robust multi-person tracking,” in IEEE/CVF Computer Vision and Pattern Recognition (CVPR), 2008.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” in IEEE/CVF Computer Vision and Pattern Recognition (CVPR), 2016.

[3] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social GAN: socially acceptable trajectories with generative adversarial networks,” CoRR, 2018.

[4] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato, “Future person localization in first-person videos,” CoRR, 2017.

[5] H. Bi, R. Zhang, T. Mao, Z. Deng, and Z. Wang, “How can i see my future? fvtraj: Using first-person view for pedestrian trajectory prediction,” in European Conference on Computer Vision (ECCV), 2020.

[6] J. Yue, D. Manocha, and H. Wang, “Human trajectory prediction via neural social physics,” 2022.

[7] J. Patrikar, B. Moon, J. Oh, and S. Scherer, “Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace,” 2021.

[8] B. Irfan, J. Kennedy, S. Lemaignan, F. Papadopoulos, E. Senft, and T. Belpaeme, “Social psychology and human-robot interaction: An uneasy marriage,” 2018.

[9] J. P. de Vicente and A. Soto, “Deepsocnav: Social navigation by imitating human behaviors,” CoRR, 2021.

[10] N. Tsoi, M. Hussein, O. Fugikawa, J. D. Zhao, and M. Vázquez, “An approach to deploy interactive robotic simulators on the web for hri experiments: Results in social robot navigation,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.

[11] A. A. E. Krishna Kumar Singh, Kayvon Fatahalian, “Krishnacam: A longitudinal, single-person, egocentric dataset for scene understanding tasks,” in IEEE Winter Conference on Applications of Computer Vision (WACV), 2016.

[12] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, Y. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Gharem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, “Ego4d: Around the World in 3,000 Hours of Egocentric Video,” in IEEE/CVF Computer Vision and Pattern Recognition (CVPR), 2022.

[13] G. Bertasius, A. Chan, and J. Shi, “Egocentric basketball motion planning from a single first-person image,” in IEEE/CVF Computer Vision and Pattern Recognition (CVPR), June 2018.

[14] A. Juliani, V. Berges, E. Vickay, Y. Gao, H. Henry, M. Mattar, and D. Lange, “Unity: A general platform for intelligent agents,” CoRR, 2018.

[15] J. Liang, L. Jiang, K. P. Murphy, T. Yu, and A. G. Hauptmann, “The garden of forking paths: Towards multi-future trajectory prediction,” CoRR, 2019.

[16] Epic Games, “Unreal engine.” [Online]. Available: https://www.unrealengine.com

[17] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in European Conference on Computer Vision (ECCV), 2016.

[18] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, “The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections,” CoRR, 2019.