

GAN Enabled Drone Trajectory Prediction

Stanford CS236G Final Project

Akram Sbaih

Department of Computer Science
Stanford University
akram@stanford.edu

Mentor **Eric Zelikman**

Department of Computer Science
Stanford University
ezelikman@cs.stanford.edu

Code available at <https://github.com/aksbaih/drone-trajectory>.

1 Introduction

Trajectory Prediction

This task is concerned with estimating the position of objects at a given point in the future. People have been interested in modeling this generic task since the beginning of science, but even more recently with the efforts to build autonomous agents that need to predict the trajectories of the environment in order to plan their own. An example of that is Pedestrian Trajectory Prediction needed when Autonomous Vehicles want to predict the trajectories of humans to avoid colliding with them.

Pedestrian Trajectory Prediction

Many datasets and benchmarks [1][2] have been introduced for the task of predicting the trajectory of pedestrians. The earlier models relied solely on recurrent encoding on the position history of the individual pedestrians to predict their future trajectory as in [3] which also tackled the social aspect of such trajectories. After that, researchers started accounting for semantics of the space which heavily improved their predictions as in [4] and [5]. Generative adversarial networks have been also used for their ability to produce multiple possible trajectories, allowing for better recall (better safe than sorry) as in [5] and [6].

Drone Racing

Drones have become popular with amateur and professional racing enthusiasts in addition to its many practical uses in the modern economy [7]. Autonomous drones are yet more challenging than automobiles because their trajectories are 3-dimensional in the air. Yet it is important for an autonomous drone to predict the behavior of other surrounding drones to avoid collision. This becomes harder and requires faster inference under racing conditions, where all drones (including the ego) are maneuvering at high velocities.

2 Dataset

Source

I'm using the UZH FPV Dataset [7] for this project. The authors of this dataset wanted to challenge the state of the art state estimation models in UAV racing by introducing highspeed maneuvers in real-world environments. This task aims to approximate the trajectory and momentum of the drone based on minimal sensory input (in this case, the grayscale view through a camera attached to the drone). The dataset consists of frames taken at 50Hz tied to a ground truth representing the momentum around each of the 6 degrees of freedom the drone has in 16 flights lasting around 100 seconds each.

Processing

I'm reusing the dataset for a different task here: I'm interested in predicting the immediate trajectory of a drone based on its location history as observed from a third-party (meaning we shouldn't assume access to the drone camera but we can observe its position history). This trajectory info is embedded



Figure 1: Example flight trajectories from the UZH FPV Dataset [7]

in the ground truth of the original dataset. Therefore, I use the dataset toolbox and my own code in the linked repo to generate the xyz location of the drone at each frame of a constant FPS and store it locally as a txt file for each flight. Since I'm going to be using transformers, I reduce the FPS to 8 instead of 50 to span larger trajectory in less memory but this is subject to change based on later experiments. This totals 11486 frames. Figure 1 shows example trajectories.

Division

Since the number of flights is not very big and each flight has a different pattern of trajectories, I reused the method of [8] where the validation set is sampled as random windows of the training set when constructing the dataloader. I'm using 64 frames from each flight for validation, making around 9% (1k frames) of all frames, but this could increase if needed. For the testing set, I left aside two flights that were considered "medium" according to [7] consisting of 1.9k frames, or 17% of the original count. This leaves 74% (8.6k frames) for training.

3 Approach

The Transformer

Transformers have shown great success with sequential tasks in NLP and other domains [9]. They also offer parallelization making them faster than recurrent approaches like LSTMs. This makes them a good fit for my task given the sequential nature of trajectories and the need for speed of inference in a drone race. The purely transformer-based approach introduced by [8] for pedestrian trajectory prediction achieved state of the art results for that task. Therefore, I'm using their approach as the baseline and backbone of my GAN-based approach.

The Adversarial Loss

The baseline transformer uses MSE loss given the spacial nature of the data. However, adversarial loss allows for our desired diversity in the output. Figure 2 shows my proposed architecture which uses the transformer as a generator that takes in the positional history $x_{i-n} \dots x_{i-1}, x_i$, concatenated with random seeds, and the temporal embedding proposed in [8] as input. While the decoder generates a trajectory y_0, y_1, \dots, y_m .

The discriminator is an encoder that takes the concatenated sequence making the candidate trajectory and gives it a fakeness score as inspired by [10]. These two components make a complete GAN that can be trained with BCE loss to generate diverse predicted trajectories.

Evaluation

I would like the predicted trajectory to be similar to the ground truth. The literature [8] uses the Mean Average Displacement MAD and the Final Average Displacement FAD. The latter of which averages the distance of the final state of the predicted trajectory from the final state of the ground truth, while the former does the same averaging over all the points of the trajectories.

Since we desire diversity in the GAN output, it is not fair to penalize it for the trajectories that don't match the ground truth. Therefore, I propose to consider only the minimum MAD and FAD over a batch of generations with diverse random vectors for the same history sequence. I could also experiment with the truncation trick to produce predictions with high fidelity and treat them the same way as the single-prediction baseline.

To measure the diversity of the GAN outputs, I propose to use a clustering algorithm (e.g. k-means)

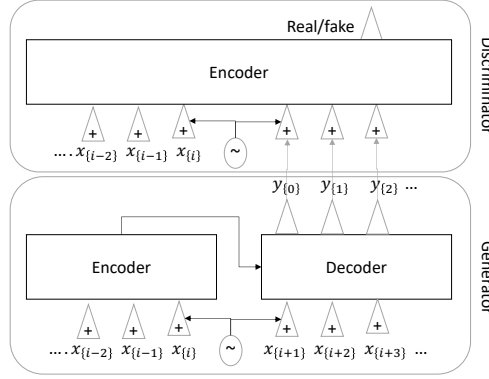


Figure 2: The proposed GAN architecture. The circle represents the temporal embedding for each timestep as proposed in [8]

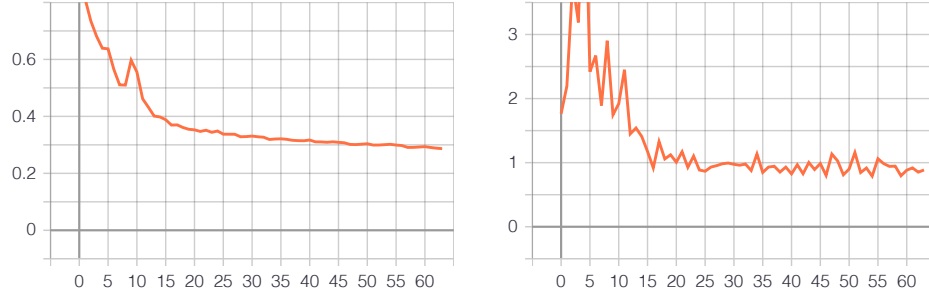


Figure 3: The x axis is the epoch. The left chart is the average loss per epoch. The right chart is the MAD on the test set at each epoch.

over its output on a batch of random vectors for the same history sequence and find the distance between these clusters. This metric is very loose because I'm not comparing the model diversity to the baseline, but it's a good indicator that there's at least some diversity in the generations.

4 Experiments and Results

The Baseline

I trained the simple transformer baseline using the same implementation linked in [8] with the training dataset for 64 epochs in 45 minutes on a Tesla K80. The batch size was 150 and the number of validation frames was 64 per flight (as described in the dataset section of this report). I also requested that the model be evaluated on the test set on every epoch for more insight given the small size of the validation set and ended up with both evaluations yielding almost identical curves with better results on the test set. Figure 3 shows the results.

5 Moving Forward

After running the baseline, I'm ready to start implementing my changes to convert the model into the GAN I proposed. After that, I will train it and run the same metrics discussed in the evaluation section. I can then compare their results and derive my conclusions on how well and diverse the GAN-based model generates.

References

- [1] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. In *European Conference on Computer Vision (ECCV)*.
- [2] Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. In *ECCV*, 2018.
- [3] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.
- [4] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints and paths to long term human trajectory forecasting, 2020.
- [5] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, S. Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints, 2018.
- [6] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks, 2018.
- [7] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.
- [8] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting, 2020.
- [9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [10] David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection, 2019.