A Solution for Pedestrian Trajectory Prediction Based on Multi-Model Fusion

Jun Yu, Xiaohua Qi, Yifan Wang, Xilong Lu, Lei Wang

University of Science and Technology of China
{harryjun, wangl}@ustc.edu.cn {xhqi,wangyfan, luxilong}mail.ustc.edu.cn

Abstract

Trajectory prediction specifically involves predicting the future position, velocity, direction, and other state information of a target object, given its historical or current motion trajectory. This discipline holds significant importance across various domains, finding extensive applications in robotics, autonomous driving, unmanned aerial vehicles (UAVs), motion analysis, and more. In the context of the challenge at hand, we are tasked with performing pedestrian trajectory prediction on the JRDB dataset, with the objective of generating predictions for pedestrians' future trajectories over a duration of 4.8 seconds at a frequency of 2.5Hz. Historically, conventional approaches have largely relied on individual trajectory prediction models, often failing to harness the complementary performance of multiple models.To address this limitation, we propose a multi-model fusion-based trajectory prediction approach. This approach amalgamates various trajectory prediction algorithms, including NSP-SFM and Y-Net, which have demonstrated strong performance on publicly available datasets. Additionally, we leverage object detection data to supplement information about pedestrian counts, resulting in a substantial reduction of the EFE Card metric to 1.293. Our approach yielded promising results in the ICCV23 Trajectory Prediction Competition.

1. Dataset

1.1. Dataset Description

Our solution heavily relies on the utilization of the JRDB dataset [1], a novel egocentric dataset collected using the social mobile manipulator, JackRabbot. This dataset constitutes a valuable resource for our research efforts, as it encompasses a comprehensive multimodal sensor dataset with detailed annotations.

The JRDB dataset comprises a wealth of data, including 64 minutes of annotated sensor information. This data encompasses various modalities, such as stereo cylindrical 360° RGB video captured at a frame rate of 15 frames per second (fps), 3D point clouds obtained from two Velodyne 16 Lidars, line 3D point clouds from two Sick Lidars, audio signals, RGB-D video captured at 30 fps, 360° spherical images captured by a fisheye camera, and encoder values from the robot's wheels. Notably, this dataset includes data from traditionally underrepresented scenarios, encompassing indoor environments and pedestrian areas, all from the ego-perspective of the robot, both when stationary and navigating.

Furthermore, the dataset has been meticulously annotated, featuring over 2.3 million bounding boxes across five individual cameras. Additionally, it includes 1.8 million associated 3D cuboids that delineate the positions of individuals in the scenes, culminating in the annotation of over 3500 time-consistent trajectories.

1.2. Data Preprocessing

During the training phase of our predictive models, we primarily employ the 3D point cloud labels for each individual within the JRDB dataset. These labels furnish the X, Y, and Z coordinates of pedestrians in the radar coordinate system. Given the specific requirements of the challenge, our objective is to predict the X and Y coordinates of pedestrians for the subsequent 4.8 seconds.

It is worth noting that the data provided by JRDB for training purposes operates at a frame rate of 15 Hz, whereas our prediction task necessitates coordinates at a lower frequency of 2.5 Hz. To address this, we perform data subsampling by extracting data every 6 frames to achieve the desired lower frequency. To maximize the utility of the data, we initiate subsampling from each of the first six frames, ensuring that each frame serves as a starting point for data extraction.

During training, we utilize data from the first 8 frames at a frequency of 2.5 Hz to predict data for the subsequent 12 frames. To maintain consistency in data processing, any data extracted that does not meet the minimum requirement of 6 frames is discarded. Additionally, we organize the data into groups, each containing 10 individuals, for the training

process.

2. Methodology

Taking into account the similarities between the JRDB dataset and the ETH [5]/UCY [3] dataset, such as their inclusion of multiple scenes, we opted to leverage several high-performing models from the ETH/UCY dataset to formulate our pedestrian trajectory prediction approach. The specific methodology is detailed as follows:

2.1. Neural Social Physics (NSP)

Neural Social Physics (NSP) [8] is a model designed to elucidate pedestrian behaviors while retaining robust data-fitting capabilities. This model amalgamates model-based and model-free approaches. Drawing inspiration from recent advances in neural differential equations, NSP introduces a novel crowd neural differentiable equation model, consisting of two integral components.

The first component entails a deterministic model expressed through a differentiable equation, wherein, unlike the social force model and its derivatives, the model parameters are learned from data, obviating the need for manual selection and fixation. This deterministic component is influenced by a dynamical system inspired by the social force model.

The second component of NSP captures intricate uncertainty inherent in motion dynamics and observations via a Variational Autoencoder. In essence, NSP constitutes a deep neural network that embeds an explicit model within its architecture, thereby fusing both model-based and model-free principles.

2.2. Y-Net

Y-Net [4] is a trajectory prediction network designed to accommodate the multimodal nature of both goal and path predictions while incorporating scene semantics. It systematically models epistemic and aleatoric uncertainties. The prediction process commences with the estimation of a probability distribution over the agent's final positions at the trajectory's end, representing epistemic uncertainty.

Subsequently, Y-Net estimates distributions over selected future waypoint positions, coupled with sampled goal points, to construct explicit probability maps encompassing all remaining trajectory positions, representing aleatoric uncertainty. The amalgamation of samples from epistemic goals and aleatoric waypoint and trajectory distributions culminates in the prediction of future trajectories.

2.3. EqMotion

In motion prediction tasks, preserving motion equivariance under Euclidean geometric transformations and ensuring invariance of agent interactions are fundamental requisites. Unfortunately, these properties are often overlooked in existing methodologies. In response, EqMotion [7] is introduced as an efficient equivariant motion prediction model with invariant interaction reasoning.

To uphold motion equivariance, EqMotion incorporates an equivariant geometric feature learning module, facilitating the acquisition of Euclidean transformable features through dedicated equivariant operations. To address agent interactions, an invariant interaction reasoning module is employed, enhancing the stability of interaction modeling. Furthermore, EqMotion introduces an invariant pattern feature learning module to extract invariant pattern features, which collaborate with the equivariant geometric features to enhance network expressiveness.

2.4. Stepwise Goal-Driven Network (SGNet)

SGNet [6] is designed to address the evolving goals of moving agents over time, providing more accurate and detailed information for future trajectory estimation. In contrast to prior approaches that exclusively model a single, long-term goal, SGNet introduces a recurrent network architecture.

This architecture comprises an encoder responsible for capturing historical information, a stepwise goal estimator that predicts successive goals into the future, and a decoder tasked with predicting future trajectories. SGNet's temporal-scale-aware modeling enables the consideration of goals at multiple temporal scales, enhancing the fidelity of trajectory predictions.

3. Test Time Augmentation

In order to harness the strengths of different predictive models and leverage their complementary advantages to enhance our final score, we applied several strategies during the testing phase.

3.1. Training Approach

For the training of human behavior trajectory prediction, we utilized the box label data from the "Labels in 3D Point Clouds" folder (referred to as 3D). Specifically, we extracted the 3D cuboid parameters: (cx, cy, cz), representing the coordinates of the cuboid center, and converted them into location coordinates (x, y) in the lidar coordinate system (measured in meters). The trajectory sequence data were then split, with 80% allocated for the training dataset and 20% for the validation dataset. This process was repeated for a total of 27 distinct scenes, resulting in 27 training datasets, each accompanied by its corresponding validation and test sets.

In pursuit of enhancing model performance, we endeavored to maximize the utilization of available data. Specifically, for each model, we conducted individual training using the training data from all 27 scenes. Recognizing the inherent variations in trajectories across different scenes, and

to further augment model performance, we subsequently fine-tuned each model using only the training data specific to each scene, while initializing the model weights from the conclusion of the initial training step.

3.2. Model Fusion

To further enhance the performance of our solution and leverage the complementary strengths of different models, we conducted a fusion of test results across multiple models, as well as employed post-processing techniques. The details are outlined below:

3.2.1 Averaging Predictions

In this approach, we compute the mean of predictions generated by all models for a particular scene. Specifically, for each model's prediction, which is a 2×12 sequence, we calculate the mean values for the x and y coordinates at each time step, yielding the fused sequence. This process can be formally described as follows:

$$\bar{x}_t = \frac{1}{N}\sum_{i=1}^{N} x_t^{(i)}, \quad \bar{y}_t = \frac{1}{N}\sum_{i=1}^{N} y_t^{(i)}, \tag{1}$$

where $\bar{x}_t$ and $\bar{y}_t$ represent the mean values of the x and y coordinates at time step t, respectively, and N is the total number of models.

Our experimental results demonstrate that this method can indeed improve the overall performance of the final model, although the improvement may not be statistically significant.

3.2.2 Optimal selection of single model data

The approach of averaging model predictions for fusion did not yield particularly favorable results. We recognized that certain models might perform poorly in specific scenarios, consequently impacting the overall prediction quality. To address this issue, we attempted to select the optimal model for each individual scene. The key question was how to determine the best-performing model.

We adopted the Evaluation on the Validation Set (EVS) score as our evaluation metric. Specifically, our approach unfolded as follows:

1. For each model, we conducted separate training using the training datasets from all 27 scenes. 2. For each scene, we performed fine-tuning of each model using only the training data specific to that scene, initializing the model weights with the results from the first training step. Subsequently, we computed the validation loss for each model on the validation dataset of that scene. This allowed us to select the model with the lowest loss for the given scene. 3. We repeated step 2 for each of the 27 scenes, resulting in the selection of the best model for each scene. These selected models were then used to infer pedestrian trajectory predictions on the respective scene's test dataset. 4. This process was iterated across all scenes, resulting in the final test results for this model fusion approach.

Mathematically, this can be described as follows:

Let M be the set of models, S be the set of scenes, D_s^{train} represent the training data for scene s, and D_s^{val} represent the validation data for scene s. The optimal model M_s^* for scene s is chosen as:

$$M_s^* = \arg\min_{M_i \in M} \text{Loss}(M_i, D_s^{train}, D_s^{val})$$

Where $\text{Loss}(M_i, D_s^{train}, D_s^{val})$ denotes the validation loss of model M_i trained on D_s^{train} and evaluated on D_s^{val}.

This model fusion approach exhibited substantial improvements compared to the simple averaging method.

3.2.3 Optimal Fusion of Multiple Model Data

We have optimized the algorithm for direct averaging and introduced a novel fusion method. In this method, we aggregate predictions generated by multiple models for a given scene. We then select three models that provide the closest predictions for the trajectories of the same individual from among the ensemble of models. Subsequently, we perform a weighted fusion of these trajectory data, with the weights being determined by the normalized losses of the corresponding models on the validation set. Our experimental results demonstrate that this approach leads to improvements in the final performance.

To elaborate, Let's assume we have n pedestrian trajectories, with each trajectory corresponding to a model, and each model achieving an accuracy on the validation set. Our goal is to select three pairs of trajectories with the smallest pairwise distances and compute a weighted average based on the accuracy of their respective models.

First, we need to define a distance metric between two trajectories. A commonly used metric is the Euclidean distance. Let's assume the i-th trajectory is denoted as T_i and the j-th trajectory is denoted as T_j. The Euclidean distance between them can be expressed as:

$$d_{ij} = \sum_{k=1}^{m} (\sqrt{(T_i[k] - T_j[k])^2})$$

Here, m is the number of time steps in the trajectory, and $T_i[k]$ represents the position of trajectory T_i at time step k.

Next, we calculate the accuracy of each model on the validation set. Let the accuracy of the i-th model be denoted as A_i.

Then, we select the three pairs of trajectories with the smallest pairwise distances.And we compute the weighted

average based on the accuracy of the selected models for the three pairs of trajectories.

By employing the Optimal Fusion of Multiple Model Data method, our solution has achieved further performance enhancement.

3.3. Correcting Predicted Human Count

Given that the evaluation metric EFE comprises two components, EFE Loc (representing average displacement error between predicted and ground-truth trajectories) and EFE Card (representing cardinality mismatch), where EFE Card penalizes deviations in predicted and actual human counts, we focused on improving EFE Card.

To address this challenge, we incorporated 3D object detection [2] results to refine our approach. Specifically, we performed 3D object detection on the final frame of the test set, considering objects with scores exceeding 0.5 as valid individuals. This strategy effectively reduced the EFE Card score, mitigating discrepancies between tracked and actual human counts in the JRDB 3D test set.

References

[1] Mahsa Ehsanpour, Fatemeh Saleh, Silvio Savarese, Ian Reid, and Hamid Rezatofighi. Jrdb-act: A large-scale dataset for spatio-temporal action, social group and activity detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[2] Yuhang He, Wentao Yu, Jie Han, Xing Wei, Xiaopeng Hong, and Yihong Gong. Know your surroundings: Panoramic multi-object tracking by multimodality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2980, 2021.

[3] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3542–3549, 2014.

[4] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[5] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[6] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022.

[7] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1420, 2023.

[8] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *European Conference on Computer Vision*, pages 376–394. Springer, 2022.