Scene-Aware LSTM-CNN-MDN for Probabilistic Pedestrian Trajectory Prediction

Malak Mahdy
Department of Computer Science
Texas A&M University-Corpus Christi
mmahdy@islander.tamucc.edu

I. Introduction

Predicting pedestrian trajectories from recent observations is a fundamental problem in autonomous driving, robotics, and crowd analysis. Given several seconds of past positions for a target pedestrian and its neighbors, the goal is to forecast future trajectories that are both accurate and socially compliant. Human motion is inherently multi-modal: a pedestrian may slow down, turn left, or turn right depending on subtle social cues and environmental context. Traditional deterministic models, such as vanilla LSTMs, often struggle to represent this uncertainty.

This project implements a trajectory prediction model combining bidirectional LSTM temporal encoding, CNN-based spatial context extraction, and a Mixture Density Network (MDN) decoder. The model outputs multiple Gaussian components over the future trajectory, enabling multi-modal probabilistic predictions. The proposed method is evaluated on several ETH/UCY subsets and compared against a constant velocity baseline and a simple LSTM baseline using ADE and FDE.

II. Related Work

Early pedestrian trajectory prediction methods relied on handcrafted dynamics and social force models. With the rise of deep learning, recurrent neural networks have become the dominant approach. Social-LSTM [1] introduced a pooling mechanism to aggregate hidden states of neighboring pedestrians, allowing the model to capture social interactions in crowded scenes.

Subsequently, generative models such as Social-GAN [2] used generative adversarial networks and a variety loss to generate socially acceptable and diverse trajectories. Graph-based approaches have also been proposed. For example, Social-BiGAT and related works [3] use graph attention mechanisms to model interactions between agents more explicitly. These methods show that attention over a spatial interaction graph is effective for learning complex social behavior.

Probabilistic approaches such as Trajectron++ [4] model multi-agent motion using a graph-structured recurrent architecture combined with a conditional variational autoencoder. Inspired by these works, this project initially explored a graph attention architecture but ultimately achieved better results with a simplified LSTM-CNN-MDN approach.

III. Proposed Method

The proposed model takes as input the past H positions of each pedestrian and outputs a distribution over the next F timesteps. The final architecture consists of three main components: a temporal encoder, a CNN-based map encoder, and an MDN decoder.

A. Temporal Encoder

Each agent's history is represented as a sequence of (x, y) positions over $H = 8$ timesteps. These sequences are passed through a bidirectional LSTM encoder, which produces a hidden representation that captures local motion patterns such as velocity and acceleration. The bidirectional nature allows the model to consider both forward and backward temporal context when encoding motion dynamics.

B. Map Encoder

The model incorporates scene context using a lightweight convolutional map encoder that processes a rasterized image patch for each agent. This encoder consists of two convolutional layers, each followed by a ReLU activation, which progressively downsample and extract spatial features from the input map. The first convolution maps the three-channel input image to 16 feature maps, and the second increases this to 32 feature maps, both using a 3×3 kernel with stride 2 and padding 1. After the convolutions, an Adaptive Average Pooling layer reduces the spatial dimensions to a single 1×1 cell, to perform global average pooling. The output is then flattened to produce a compact 32-dimensional map embedding. This embedding captures the local scene structure around the agent and is combined with the temporal features before being projected into the shared latent space for trajectory decoding.

C. Mixture Density Network Decoder

To capture the fact that future human trajectories are inherently uncertain and may follow multiple plausible paths, the model uses a Mixture Density Network (MDN) decoder. Instead of producing a single predicted future, the MDN outputs a mixture of K Gaussian components. For each component, the network predicts:

- a mean trajectory μ_k,
- a standard deviation σ_k representing uncertainty,
- and a mixture weight π_k indicating the likelihood of that component.

These components together define a probabilistic mixture model over future trajectories. The learning objective is based on the negative log-likelihood (NLL) of the ground-truth trajectory under this mixture distribution:

$$\mathcal{L}_{\text{MDN}} = -\log\left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(y \mid \mu_k, \sigma_k)\right),$$

where y denotes the observed future positions.

This expression computes how probable the true future trajectory is according to the model's predicted mixture. If the model assigns high probability to the actual trajectory (e.g., one of the modes closely matches the ground truth), the loss will be low. Conversely, if all predicted modes place low probability on the true future, the loss increases. This formulation enables the model to represent uncertainty and multi-modality in human motion, allowing multiple distinct future paths to be learned within a single probabilistic framework.

D. Graph Attention Network: Initial Implementation

An initial implementation included a Graph Attention Network (GAT) layer to model social interactions between pedestrians. At each scene frame, pedestrians were treated as nodes in a graph, with the GAT computing attention scores between neighboring nodes. However, this implementation contained a critical fault: the GAT layer computed attention across batch samples rather than within-scene pedestrians. This effectively randomized the social modeling component and significantly degraded performance. After identifying this issue, the GAT module was removed, resulting in the simplified LSTM-CNN-MDN architecture that achieved superior results.

IV. Experiments

A. Datasets

The proposed approach is evaluated on five subsets from the ETH/UCY benchmark: ETH-Hotel, ETH-University, Zara01, Zara02, and Students [5]. Each dataset contains trajectories of outdoor or semi-crowded scenes. The data are provided as world-space (x, y) coordinates for each pedestrian over time.

For each agent, the $H = 8$ historical timesteps is used, and the $F = 12$ future timesteps is predicted. All five CSV files are combined into a single dataset. A deterministic 80/20 split using a fixed random seed in PyTorch is implemented to ensure that the training and evaluation sets are disjoint and reproducible.

B. Training Details

The LSTM-CNN-MDN model was trained for approximately 100 epochs (5 training sessions of 20 epochs each) using the Adam optimizer with a learning rate of 1×10^{-3} and batch size of 32. Training was conducted on CPU with the fixed random seed ensuring reproducible splits. The iterative training approach allowed for debugging and refinement of the model architecture during development.

C. Baselines

To contextualize the performance of the LSTM-CNN-MDN model, two baselines are implemented:

1) Constant Velocity (CV): The constant velocity baseline does not require learning. It estimates the last observed velocity from the historical positions and linearly extrapolates it into the future. Despite its simplicity, this method can be surprisingly strong for short prediction horizons and mostly linear motion.

2) LSTM Baseline: The LSTM baseline uses a single-agent LSTM encoder-decoder architecture. It takes the past H historical positions as input and predicts a deterministic sequence of F future positions. This baseline does not include map encoding or mixture density outputs and is trained with a mean squared error (MSE) loss. It serves to demonstrate the value added by the CNN map encoder and probabilistic MDN decoder.

D. Evaluation Metrics

Two standard metrics from the trajectory prediction literature is adapted [6]:

- **Average Displacement Error (ADE):** The mean L2 distance between predicted and ground-truth positions over all future timesteps.
- **Final Displacement Error (FDE):** The L2 distance between the predicted and ground-truth positions at the final future timestep.

Lower ADE and FDE indicate better performance, with ADE reflecting overall trajectory quality and FDE reflecting destination accuracy.

E. Evaluation Results

Table I shows the initial results with the GAT implementation, while Table II presents the final results after removing the GAT module.

TABLE I
INITIAL RESULTS WITH GAT IMPLEMENTATION ON THE COMBINED ETH/UCY TEST SPLIT.

Model	ADE ↓	FDE ↓
ST-GAT + MDN	4.37	4.96
Constant Velocity	0.50	1.10
LSTM Baseline	0.59	1.11

TABLE II
FINAL RESULTS WITH LSTM-CNN-MDN ARCHITECTURE ON THE COMBINED ETH/UCY TEST SPLIT.

Model	ADE ↓	FDE ↓
LSTM-CNN-MDN	**0.48**	**1.00**
Constant Velocity	0.50	1.10
LSTM Baseline	0.59	1.11

After identifying and removing the GAT implementation, the LSTM-CNN-MDN model achieves the best performance across both metrics, outperforming both baselines. The model achieves an ADE of 0.48 and FDE of 1.00, representing approximately 4% and 9% improvement over the constant velocity baseline, respectively. This demonstrates that the

combination of bidirectional LSTM temporal encoding, CNN-based spatial context, and probabilistic MDN outputs effectively captures the patterns in pedestrian trajectories on this dataset.

The strong performance compared to baselines suggests that the model successfully leverages both temporal dynamics and spatial scene context. The MDN decoder allows the model to represent uncertainty in predictions while still achieving accurate single-mode outputs when evaluated with standard metrics.

V. Conclusion

This project implemented a pedestrian trajectory prediction model combining bidirectional LSTM encoding, CNN-based map features, and a Mixture Density Network decoder. Initial attempts to incorporate graph attention for social interaction modeling revealed implementation challenges, leading to a simplified but more effective architecture. The final LSTM-CNN-MDN model was evaluated on the ETH/UCY datasets and compared against constant velocity and LSTM baselines using ADE and FDE metrics.

The final model achieves superior performance on both metrics, demonstrating that the combination of temporal encoding, spatial context, and probabilistic outputs is effective for this prediction task. The iterative development process, including identifying and resolving the GAT implementation, provided valuable experience in debugging deep learning architectures and understanding the trade-offs between model complexity and performance.

Future work could include: properly implementing graph attention with correct within-scene batching for social interaction modeling, exploring more sophisticated scene encoders such as semantic segmentation features, and evaluating on additional datasets with more complex multi-modal scenarios where the probabilistic nature of the MDN decoder would provide greater advantages.

References

[1] A. Alia, M. Chraibi, and A. Seyfried, "Social lstm with dynamic occupancy modeling for realistic pedestrian trajectory prediction," *arXiv preprint arXiv:2511.09735*, 2025.

[2] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, no. CONF, 2018.

[3] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, S. H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *CoRR*, vol. abs/1907.03395, 2019. [Online]. Available: http://arxiv.org/abs/1907.03395

[4] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," *CoRR*, vol. abs/2001.03093, 2020. [Online]. Available: http://arxiv.org/abs/2001.03093

[5] H. Wei, "Dataextraction: Ucy and eth dataset preprocessing scripts," https://github.com/erichhhho/DataExtraction, 2017, gitHub repository, MATLAB scripts for UCY/ETH pedestrian datasets.

[6] J. B. Fernández, "Error metrics for trajectory prediction accuracy," https://jaimefernandezdcu.wordpress.com/2019/02/07/error-metrics-for-trajectory-prediction-accuracy/, 2019, blog post.