Multi-Agent Tensor Fusion for Contextual Trajectory Prediction

Tianyang Zhao[1], Yifei Xu[2], Mathew Monfort[3], Wongun Choi[4], Chris Baker[4], Yibiao Zhao[4], Yizhou Wang[5,6,7], Ying Nian Wu[2]

[1]*Peking University,* [2]*University of California, Los Angeles,* [3]*MIT CSAIL,* [4]*ISEE,* [5]*Computer Science Dept., Peking University,* [6]*Peng Cheng Lab,* [7]*Deepwise AI Lab*

{zhaotianyang, yizhou.wang}@pku.edu.cn, fei960922@ucla.edu, mmonfort@mit.edu, {wchoi, chrisbaker, yz}@isee.ai, ywu@stat.ucla.edu

Abstract

Accurate prediction of others' trajectories is essential for autonomous driving. Trajectory prediction is challenging because it requires reasoning about agents' past movements, social interaction among varying numbers and kinds of agents, constraints from the scene context, and the stochastic nature of human behaviors. Our approach models these interactions and constraints jointly within a novel Multi-Agent Tensor Fusion (MATF) network. Specifically, the model encodes multiple agents' past trajectories and the scene context into a Multi-Agent Tensor, then applies convolutional fusion to model multiagent interactions while retaining the spatial structure of agents and the scene context. The model decodes recurrently to multiple agents' future trajectories, using adversarial loss to learn stochastic predictions. Experiments on both autonomous driving and pedestrian crowd datasets show that the model achieves state-of-the-art prediction accuracy.

1. Introduction

Accurate prediction of the trajectories of nearby vehicles and pedestrians is essential for safe autonomous driving. Proactive driving needs precise prediction of future events to get prepared for the unknown future. However, the prediction task is challenging by its nature, because these agents' motions depend on the scene context, as well as their goals and social interactions with other agents. Also, predictions must generalize to new situations, where the number and configuration of other agents are not fixed in advance.

For neural-net-based approaches, modeling social interaction among varying numbers of agents is inherently difficult. This is because human interactive behavior is complex and stochastic, and also because neural nets prefer fixed shapes of parameters instead of varying ones. Thus, prediction models have to handle varying numbers of agents by parameter sharing among units for each agent, and by aggregation functions insensible to the varying number of agents to fuse information from different agents.

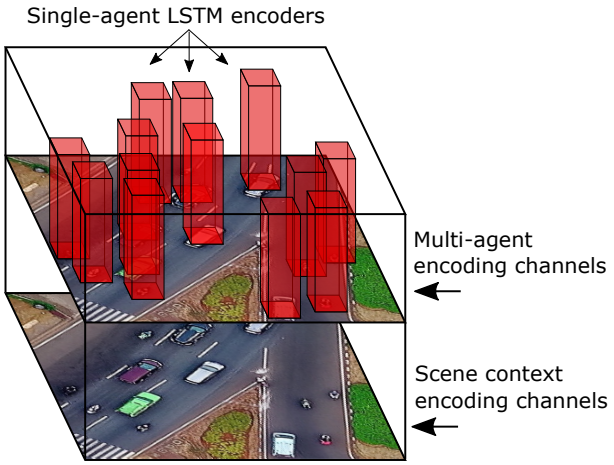

Figure 1. Illustration of the Multi-Agent Tensor encoding. It is a top-down view spatial feature map of the scene context and multiple agents, including agent channels (above) and context channels (below). Agents' feature vectors (red) output from Single-Agent LSTM encoders are placed spatially w.r.t. agents' coordinates to form the agent channels. The agent channels are aligned spatially with the context channels (a context feature map) output from scene context encoding layers to retain the spatial structure.

Generally, previous NN-based works model multiagent interaction either in an agent-centric way or in a spatial-centric way. Agent-centric approaches apply aggregation functions on multiple agents' feature vectors, while spatial-centric approaches operate directly on top-down images of the scene. We absorb merits from both sides by proposing a novel NN-based Multi-Agent Tensor Fusion (MATF)

for the prediction problem. The proposed model jointly predicts future trajectories of varying numbers of agents by modeling social interaction among them and constraints from the scene context addressing both the agent-centric and the spatial-centric aspects. Specifically, to achieve this, our key idea is to construct a Multi-Agent Tensor retaining the spatial structure of the agents and scene contexts, and to model the interactions and constraints by running shared fully-convolutional layers upon the tensor.

We conduct experiments on both driving datasets and pedestrian crowd datasets. Experimental results are reported on the publicly available NGSIM driving dataset [7], Stanford Drone pedestrian crowd dataset [25], ETH-UCY crowd datasets [21, 27], and a private recently-collected highway driving dataset. Quantitative and qualitative ablative experiments are conducted to show the contribution of each part of the model, and quantitative comparisons with recent approaches show that the proposed approach achieves state-of-the-art accuracy in both autonomous driving and pedestrian trajectory prediction.

2. Related Work

Traditional methods for predicting or classifying trajectories model various kinds of interactions and constraints by hand-crafted features or cost functions [3, 5, 6, 8, 15, 22, 32]. Inverse optimal control methods also use hand-crafted cost features, but use learning to estimate linear weights to rationalize trajectories which are assumed to be generated by optimal control [18]. Recent data-driven approaches based on deep networks [1, 4, 9, 10, 13, 19, 20, 24, 28, 29, 31] outperform traditional approaches. Most of these works contributes either on modeling constraints from the scene context [29] or on modeling social interactions among multiple agents [1, 9, 10, 13, 31]. A small fraction of these works jointly consider both aspects [4, 20, 28].

Many of these NN-based approaches have made great contribution to address information sharing among multiple agents by applying aggregation functions on multiple agents' feature vectors output from recurrent units. Social LSTM [1] is a pioneering one among these agent-centric approaches. Social LSTM runs max pooling over state vectors of nearby agents within a predefined distance range. In this way, it does not model social interaction between agents far away. Social GAN [13] contributes by exploring a new pooling mechanism over all the agents involved in a scene globally, and by using GAN to model the stochastic nature of human behaviors [12]. Although these kinds of max pooling aggregation functions handle varying number of agents well, permutation invariant functions may discard information when input agents lose their uniqueness, as pointed out by [28]. In contrast, Social Attention [31] and Sophie [28] address the heterogeneity of social interaction among different agents by attention mechanisms [2, 30] and spatial-temporal graphs [17]. Attention mechanisms encode which other agents are most important to focus on when predicting the trajectory of a given agent. However, attention-based approaches are very sensitive to the number of agents included, with $O(n^2)$ computational complexity to predict n agents. In contrast, our approach address this heterogeneity while still operates within an $O(n)$ computational complexity.

The agent-centric approaches above do not make use of spatial relationships among agents directly. As an alternative, spatial-centric approaches contributes by retaining or partially retaining the spatial structures of agents and the scene context during their operation process. Social Convolutional Pooling [9] contributes by partially retaining this spatial structure. However, this approach does not encode the scene context, and the trajectory of only one agent can be predicted with each forward pass – potentially slow for real-time trajectory prediction of multiple agents. Chauffeur Net [4] also contributes as a novel method to retain spatial structure of agents and the scene by directly operating on the spatial feature map of gents and the scene context. In this approach, agents are represented as bounding boxes and does not have independent recurrent encoding units. In contrast, our model encodes multiagents' feature vectors via recurrent units while simultaneously retains the spatial structure of agents and the scene throughout the reasoning process.

Many data-driven approaches learn to predict deterministic futures of agents by minimizing reconstruction loss [1, 29]. However, human behavior is inherently stochastic. Recent approaches address this by predicting a distribution over future trajectories either by combining Variational Auto-Encoders [11] and Inverse Optimal Control [20], or by conditional Generative Adversarial Nets [13, 28]. Other approaches predict future trajectories by conditioning on possible maneuver classes, to predict a set of possible trajectories instead of a deterministic one [9, 10]. GAIL-GRU [19] uses generative adversarial imitation learning [16] to learn to imitate human expert driving behaviors. R2P2 [24] proposes a novel cost function to encourage enhancement in both precision and diversity of the leanred prediction.

3. Method

In this section, we propose a novel approach Multi-Agent Tensor Fusion (MATF) to trajectory prediction by jointly modeling social interaction among multiple agents and constraints from the scene context. Given a scene containing a static context and multiple dynamic interacting agents, the top-down view representation of the static scene context is denoted as c, which can be either a segmented image containing all static objects, or a birds-eye view raw image. The number of agents n varies in different scenes. Past trajectories of the agents are denoted $\{x_1, x_2, .., x_n\}$,

where x_i is the i-th agent's history over past time period T. The objective is to predict the future trajectories of these agents jointly $\{y_1, y_2, .., y_n\}$ over future period T'. Figure 2 gives an overall illustration of the MATF architecture.

3.1. MATF Encoding

There are two paralleled encoding streams in MATF. One encodes the static scene context image c using CNNs, and another encodes past trajectories of each individual agent x_i independently using single agent LSTM Encoders. Each LSTM Encoder shares the same set of parameters, so the architecture is insensitive to varying numbers of agents in different scenes. The output of the scene context encoder CNNs is a scaled feature map c' retaining the spatial structure of the top-down scene context image, and the outputs of the LSTM encoders are 1-d agent state vectors $\{x'_1, x'_2, .., x'_n\}$ without temporal structure.

Next, the two encoding streams are concatenated spatially into a top-down view discrete spatial Multi-Agent Tensor in the following way: Agent encodings $\{x'_1, x'_2, .., x'_n\}$ are placed into one discrete top-down view spatial tensor, which is initialized as 0 and is of the same shape (width and height) as the encoded scene image c'. The dimension axis of the encodings fits into the channel axis of the tensor as shown in Figure 2. The agent encodings are placed into the spatial tensor with respect to their positions at the last time stamp of their past trajectories, respectively. This tensor is then concatenated with the encoded scene image in the channel dimension to get a combined spatial tensor. If multiple agents are placed into the same cell in the spatial tensor due to discretization, element-wise max pooling is performed. This spatial fusing method ensures that the architecture is not sensitive to varying numbers of agents in different scenes.

3.2. MATF Decoding

This Multi-Agent Tensor is then fed into fully convolutional layers, which learn to represent interactions among multiple agents and between agents and the scene context, while retaining spatial locality. Specifically, these layers operate at multiple spatial resolution scale levels by adopting U-Net-like architectures [26] to model interaction at different spatial scales. The output feature map of this fusion model c'' shares exactly the same shape with c' in width and height to retain the spatial structure.

Subsequently, agent-specific representations with fused interaction features for each agent $\{x''_1, x''_2, .., x''_n\}$ are sliced out according to their coordinates from the fused spatial tensor output c''. These agent-specific representations are then added as a residual [14] to the original encoded agent vectors to form fused agent vectors $\{x'_1 + x''_1, x'_2 + x''_2, .., x'_n + x''_n\}$, which encode all the information from the past trajectories of the agents themselves, the static scene context, and

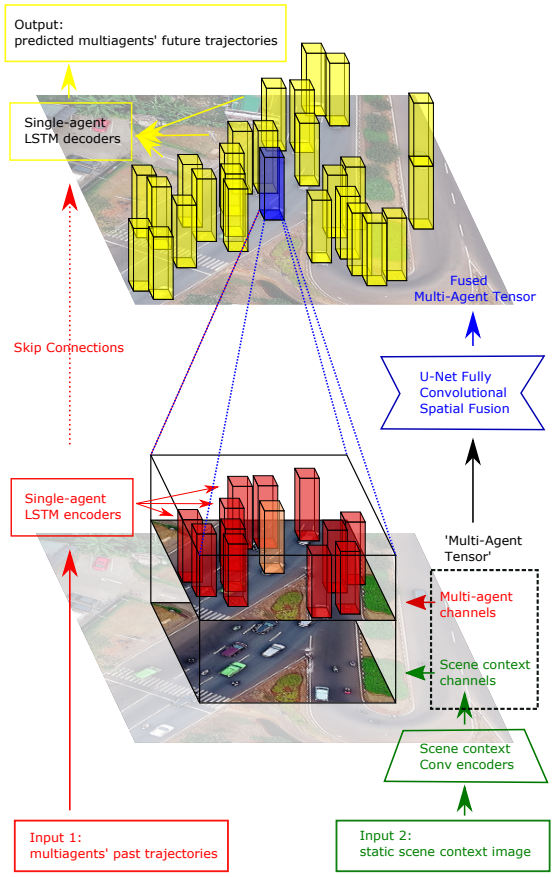

Figure 2. Illustration of the Multi-Agent Tensor Fusion (MATF) architecture. The inputs are n agent past trajectories $\{x_1, x_2, .., x_n\}$, and a top-down view static scene context image c. c and each x_i are encoded independently through convolutional and recurrent encoding streams, the green and red streams respectively. Then, the top-down view encoded scene context c' and encoded agent vectors $\{x'_1, x'_2, .., x'_n\}$ from the two streams are aligned spatially together to form a Multi-Agent Tensor. The big black box shows a part of the tensor surrounding the orange agent. Subsequently, U-Net like fully convolutional spatial fusion layers are applied on top of the Multi-Agent Tensor to reason about the interactions while retaining spatial locality to output a fused Multi-Agent Tensor c'', the feature map above. Fused vectors for each agent $\{x''_1, x''_2, .., x''_n\}$, the yellow and blue boxes above, are then sliced out from c'', which contain the interaction, history, and constraint features for the corresponding agent. Note that because we run shared convolution, corresponding fused vectors of all the agents are generated in one forward path. For instance, x''_i, the small blue agent box above, contains fused information from all the agents and scene information nearby agent i within the receptive field of the convolutional layers, the big black box below in this case. These fused vectors are then added to the original encoded vectors of the corresponding agents as residuals and get decoded independently to future trajectory predictions. The whole architecture is fully differentiable and is trained end-to-end.

the interaction features among multiple agents. In this way, our approach allows each agent to get a different social and context embedding focused on itself. Furthermore, the fusion model gets these embeddings for multiple agents using shared feature extractors instead of operating n times for n agents.

Finally, for each agent in the scene, its fused vector $x_i' + x_i''$ is decoded to future trajectory prediction $\hat{y}_i$ by LSTM Decoders. Similar to the encoders for each agent, parameters are shared to guarantee that the network can generalize well when the number of agents in the scene varies.

The whole architecture is fully differentiable and is trained end-to-end to minimize reconstruction loss between $\{\hat{y}_1, \hat{y}_2, .., \hat{y}_n\}$ and $\{y_1, y_2, .., y_n\}$.

3.3. Adversarial Loss

To further learn stochastic predictions, we use conditional generative adversarial training [12, 23] upon MATF.

GANs consist of two networks, a generator G and a discriminator D competing against each other. G learns the distribution of the data and generates samples, while D learns to distinguish the feasibility or infeasibility of the generated samples. These networks are simultaneously trained in a two player min-max game framework.

In our setting, we use a conditional G to generate future trajectories of multiple agents jointly, conditioning on all the agents' past trajectories, the static scene context, and random noise input to create stochastic outputs. Simultaneously, we use D to distinguish whether the generated paths are real (ground truth) or fake (generated). Both G and D share exactly the same architecture in their encoding parts with the deterministic model presented in Section 3.1, to reason about static scene context and interaction among multiple agents spatially. Both G and D are initialized with parameters from the trained deterministic model introduced in previous subsections. Detailed architectures and losses are introduced below.

Generator (G) G observes past trajectories of all the agents in a given scene $\{x_1, x_2, .., x_n\}$, and the static scene context c. It jointly outputs the predicted future trajectories $\{\hat{y}_1, \hat{y}_2, .., \hat{y}_n\}$ by decoding the fused agent vectors $\{x_1' + x_1'', x_2' + x_2'', .., x_n' + x_n''\}$ in Section 3.2, concatenated with Gaussian white noise vector z. Its architecture is exactly the same as presented in previous subsections, except that in the deterministic model, the fused encoding for a given agent $x_i' + x_i''$ is concatenated with $z = 0$ vector to decode into its future trajectory; while in G, z is sampled from a Gaussian distribution.

Discriminator (D) D observes either all generated or all ground truth past and future trajectories of the agents $\{x_1, x_2, .., x_n, \hat{y}_2, .., \hat{y}_n\}$ or $\{x_1, x_2, .., x_n, y_1, y_2, .., y_n\}$ in a given static scene context c. It outputs real or fake labels for the future trajectory of each agent in the scene $\{\hat{l}_1, \hat{l}_2, .., \hat{l}_n\}$. It shares nearly the same architecture as presented in previous subsections, except for the following differences: (1) Its single agent LSTM encoders take in past and future trajectories as input instead of just past trajectories; (2) As a classifier, it does not use the LSTM decoders to decode $x_i' + x_i''$ to future trajectories. Instead, the agent fused encodings $x_i' + x_i''$ are fed into fully connected layers to be classified as a real or a fake trajectory of agent i: $\hat{l}_i$.

Losses To train the Multi-Agent Spatial GAN, we use the following losses:

$$W^* = \arg\min_W \mathbb{E}_{scene}[\mathcal{L}_{GAN}(\{\hat{l}_i, l_i\}_{i \in scene}) + \lambda \sum\nolimits_{i \in scene} L_{L2/L1}(\hat{y}_i, y_i)], \tag{1}$$

where W is the set of weights of the model and λ is a regularizer between adversarial loss and reconstruction loss. The adversarial loss L_{GAN} is:

$$\mathcal{L}_{GAN}(\{\hat{l}_i, l_i\}_{i \in scene}) = \min_G \max_D \sum_{i \in scene} \mathbb{E}_{traj_i \sim p(y_i)}[l_i \log \hat{l}_i] + \mathbb{E}_{traj_i \sim p(\hat{y}_i)}[(1 - l_i) \log(1 - \hat{l}_i)]. \tag{2}$$

4. Experiments

In the Experiments and Results sections, we evaluate our model on both driving datasets [7] and a pedestrian crowd dataset [25]. We construct different variants of our models for ablative studies, and compare with state-of-the-art alternative methods [1, 8, 9, 13, 15, 19] quantitatively. Qualitative results are also presented for further analysis.

4.1. Datasets

We use the publicly available NGSIM US101 dataset [7], the publicly available Stanford Drone Dataset [25], and a recently collected highway driving dataset for training and evaluation.

NGSIM. NGSIM US101 dataset [7] is a driving dataset consisting of trajectories of real freeway traffic over a time span of 45 minutes. It is recorded by fixed birds-eye view cameras placed over a 640-meter span of US101. Trajectories of all the vehicles traveling through the area within this 45 minutes are annotated. The dataset consists of various traffic conditions (mild, moderate and congested), and contains around 6k vehicles in total.

Autonomous Driving. NGSIM contains rich vehicle-vehicle interactions. However, the recorded highway span is quite straight, so minimal agent-scene interaction is observed. As an alternative, we analyze a large-scale dataset gathered by an autonomous vehicle during highway driving, including a several-mile stretch of highway, with curved lanes, highway exits and entrances. Ablative studies are conducted for this dataset to show our model's ability to

model agent-scene and agent-agent interactions, respectively.

Stanford Drone. Stanford Drone dataset is a large-scale pedestrian crowd dataset [25] consisting of a top-down view of 20 unique scenes in which pedestrians, bicyclists, skateboarders, carts, cars and buses navigate on a university campus. Raw, static scene context images are provided from a top-down view, and coordinates of multiple agents' trajectories are provided in pixels. These scenes contain rich human-human interactions, often in the form of high density crowds, and diverse physical landmarks such as buildings and roundabouts that must be avoided. We crop the videos into around 16k scenes, and we use the standard test set for quantitative evaluation. Also note that some scenes from the standard training set are not used for our training process, but leaved for qualitative evaluation instead.

4.2. Baseline Models

We compare quantitatively with the following published baseline approaches, and we also construct a set of variants of our model for ablative studies.

Baselines:

CV: A simple constant velocity Kalman filter.

C-VGMM + VIM [8]: A recent vehicle interaction approach based on variational Gaussian mixture models with Markov random fields.

GAIL-GRU [19]: State-of-the-art vehicle trajectory prediction model. It uses GRU-based GAIL [16] to generate the dynamics of a bicycle model of vehicle motion. In this approach, trajectories are generated one vehicle at a time, and the model assumes access to the ground truth trajectories of nearby vehicles over the prediction horizon, which other approaches, including our, do not.

Social Conv [9]: State-of-the-art vehicle trajectory prediction approach based on maneuver modeling. This model uses extra supervision of maneuver labels for a two-step training procedure to model the latent distribution space for future trajectories. They also evaluate on NGSIM [7], so we design our experimental setting identically and directly compare with their reported quantitative results.

Social Force [15]: Pioneering traditional feature-based pedestrian trajectory prediction model.

Social LSTM [1]: State-of-the-art deterministic pedestrian trajectory prediction model.

Social GAN [13]: State-of-the-art stochastic pedestrian trajectory prediction model. They report the error of the best sample among N generated samples for stochastic prediction evaluation. We also report this for fair comparison with our stochastic models.

Desire [20]: State-of-the-art stochastic pedestrian trajectory prediction model. They use Variational Auto-Encoders [11] and Inverse Optimal Control to generate and rank trajectories.

Sophie [28]: A complex, state-of-the-art stochastic attention-based prediction model. They use the same evaluation metrics as Social GAN [13], and they evaluate on the same pedestrian crowd dataset [25] as we do here. We design our experimental setting identically, and compare with their reported quantitative results.

Variants:

LSTM: A simple LSTM Encoder-Decoder as introduced in Section 4.1.

Single Agent Scene: The model shares exactly the same architecture as introduced in Section 3.2, except that it only takes in one agent history x_i with scene representation c and outputs only $\hat{y}_i$ each time, so the model reasons about scene-agent interaction, but is completely unaware of multi-agent interaction.

Multi Agent: The model has the same details as the described model in Section 3.2, except that scene representation c is not provided as input. The model only reasons about multi-agent interaction absent from scene context information.

Multi Agent Scene: The introduced deterministic model in Section 3.2.

Individual D: The stochastic model which shares the same architecture with the GAN introduced in Section 3.3, except that its D only takes in each agent's individual encoding x_i' instead of $x_i' + x_i''$ for classification.

Joint D: The stochastic model introduced in Section 3.3. Similar to Social GAN [13], we sample N times and report the best trajectory in the L2 sense for fair comparison with stochastic models, with $N = 3$ in Section 5.1, and $N = 20$ as adopted by [13] in Section 5.2.

See Supplementary Materials for implementation details.

5. Results

5.1. Driving Datasets

We conducted experiments on the publicly available NGSIM US101 [7] dataset. We adopt the same experimental setting as reported in [9]: We split the trajectories into segments of 8s, and all agents appearing in the 640-meter span are considered in the reasoning and prediction process. We use 3s of trajectory history and a 5s prediction horizon. LSTMs operate at 0.2s per timestep. As in [9], we report the RMSE in meters with respect to each timestep t within the prediction horizon:

$$RMSE(t) = \sqrt{\frac{1}{n} \sum_{i=1,2,..,n} ((\hat{x}_{it} - x_{it})^2 + (\hat{y}_{it} - y_{it})^2)}, \quad (3)$$

where n is the total number of agents in the validation set, x_{it} denotes the x coordinate of the i-th car in the dataset at

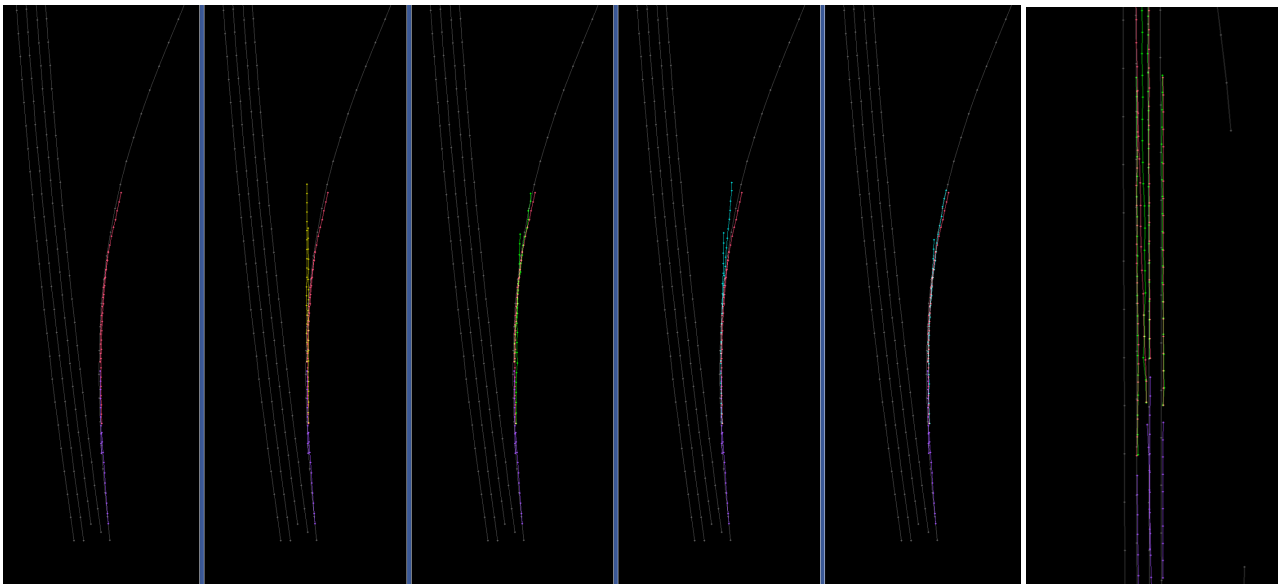

Figure 3. Qualitative results from Autonomous Driving dataset. The left 5 figures are extracted from one scene for ablative study. In these 5 figures, the grey lines indicate lane centers, purple past trajectories, red ground truth future trajectories. From left to right shows predicted results of ground truth (red), *LSTM* (yellow), *Single Agent Scene* (green), *Multi Agent* (blue), and *Multi Agent Scene* (blue). The right most figure shows an interesting prediction result of the *Multi Agent* model. In this case, the following car in the middle is taking over the leading car. All results are from validation set.

future timestamp t, and y_{it} the y coordinate respectively.

RMSE	1s	2s	3s	4s	5s
CV [9]	0.73	1.78	3.13	4.78	6.68
Our LSTM	0.66	1.62	2.94	4.63	6.63
C-VGMM + VIM [8]	0.66	1.56	2.75	4.24	5.99
GAIL-GRU [19]	0.69	1.51	2.55	3.65	4.71
Our Multi Agent	0.67	1.51	2.51	3.71	5.12
Social Conv [10]	**0.61**	**1.27**	2.09	3.10	4.37
Our Joint D	0.66	1.34	**2.08**	**2.97**	**4.13**

Table 1. Quantitative results on NGSIM [7] dataset. RMSEs in meters with respect to each future timestep in the prediction horizon are reported.

Quantitative results are shown in Table 1. This result shows that our deterministic model *Multi Agent* achieves the state-of-the-art accuracy as other deterministic models *C-VGMM + VIM, GAIL-GRU*. Note that *GAIL-GRU* has access to the future gound-truth of other agents while predicting a given agents, but we do not have access to it. The result also shows compared with approaches that capture distribution or maneuvers of future trajectories *Social Conv*, our stochastic model *Joint D* also performs at the state-of-the-art level. Note that *Social Conv* have access to auxiliary supervision of maneuver labels, while we do not encourage this. Lanes in NGSIM dataset are quite straight, so little agent-scene interaction is included, and our *Multi Agent Scene* model does not outperforms *Multi Agent*.

We also explore our Autonomous Driving dataset, where more curved lanes and more complex static scene contexts are included. We report the MAE in meters with respect to each timestep t within the prediction horizon:

$$MAE(t) = \frac{1}{n} \sum_{i=1,2,..,n} \sqrt{(\hat{x}_{it} - x_{it})^2 + (\hat{y}_{it} - y_{it})^2} \quad (4)$$

Quantitative ablative results are shown in Figure 3, and some interesting qualitative examples are shown in Figure 2. Quantitative results show that both *Single Agent Scene* and *Multi Agent* outperforms *LSTM* baseline, and that *Multi Agent* consistently outperforms *Single Agent Scene* and *Multi Agent*, and comparison between *Multi Agent* and *Single Agent Scene* shows that the former one performs better in the short term while the latter one performs better in the longer term. Qualitative results give reason to these observations.

Left 5 figures in Figure 2 shows a typical agent-scene interaction scenario. In this scene, the ground truth future of the two cars are on a exit of a highway driving along a curved lane. Without multi-agent or scene information, the predicted results from *LSTM* learns a strategy of going straight out of the lane; while all the other 3 models follow lane in different accuracy levels. With scene context information, *Single Agent Scene* and *Multi Agent Scene* learn to

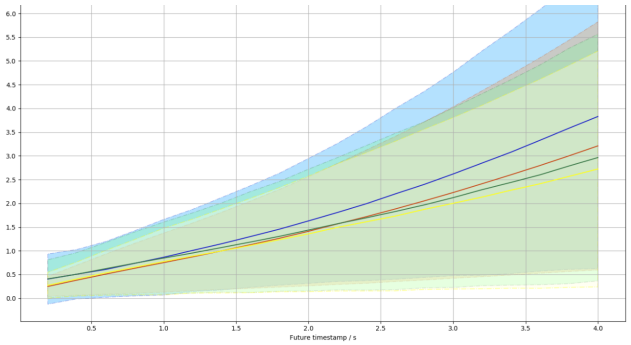

Figure 4. Quantitative results on Autonomous Driving dataset. MAEs in meters with respect to each future timestep in the prediction horizon are reported. Blue line is the evaluation result of *LSTM*, red for *Multi Agent*, green for *Single Agent Scene*, and yellow for *Multi Agent Scene*.

follow the lane as expected. However, we notice that *Multi Agent* can also follow the lane in a shorter future although finally fails in distant future. This may because the following car can follow the leading car's past trajectory which is an approximation to the lane shape, and because the followed car can reason the lane shape by the relative position of multiple cars and follow an 'imaginery' lane. This lane may be correct in the near future, but may diverge from reality in the distant future, where curvature of the lane changes. This explains why *Single Agent Scene* outperforms *Multi Agent* in the distant future.

While, the rightmost figure shows a typical agent-agent interaction scenario. The leading car in the middle is traveling slower than the following car, and the *Multi Agent* model successfully predicts a taking-over of the leading car, and results in its changing lane behavior. This kind of changing lane behavior promote the prediction accuracy from the start of future time span, and this explains why *Multi Agent* outperforms *Single Agent scene* in the near future.

Overall from these ablative studies, we conclude that our spatial fusion model successfully models agent-agent and agent-scene interaction. More specifically, the scene fusion model learns constraints from scene context, and the multi-agent model learns multi-agent interaction. These interaction have different types, including following behaviors, spatial reasoning behaviors, and competing behaviors.

5.2. Pedestrian Dataset

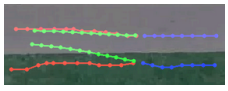

Figure 5. A collision avoidance example predicted by *Multi Agent* in Stanford Drone [25] dataset. Refer to Figure 5 for legends.

We also explore a pedestrian dataset, the publicly available Stanford Drone [25] dataset. We adopt the same experimental setting as reported in [28]: We split the trajectories into segments of 8s, and all agents appearing in the scene are considered in the reasoning and prediction process. We use 3.2s of trajectory history and a 4.8s prediction horizon. LSTMs operate at 0.4s per timestep. As in [25], we report the ADE and FDE in pixels with respect to each timestep t within the prediction horizon:

$$
\begin{aligned}
ADE(i) &= \frac{1}{t} \sum_{j=1,2,..,T'} \sqrt{(\hat{x}_{ij} - x_{ij})^2 + (\hat{y}_{ij} - y_{ij})^2} \\
ADE &= \frac{1}{n} \sum_{i=1,2,..,n} ADE(i) \\
FDE(i) &= \sqrt{(\hat{x}_{iT'} - x_{iT'})^2 + (\hat{y}_{iT'} - y_{iT'})^2} \\
FDE &= \frac{1}{n} \sum_{i=1,2,..,n} FDE(i)
\end{aligned}
\tag{5}
$$

where n is the total number of agents in the validation set, x_{ij} denotes the x coordinate of the i-th car in the dataset at future timestamp j, and y_{ij} the y coordinate respectively, T' denotes final future timestamp. Quantitative results show below in Table 2. Note that the 5 models on the top are deterministic ones, while the bottom 5 capture distribution. The proposed $MultiAgentScene$ outperforms other deterministic model in ADE, and the propsed $JointD$ performs at the state-of-the-art level.

Method	ADE	FDE (4.8s)
Our LSTM	37.35	77.13
Social Force [15]	36.38	58.14
Social LSTM [1]	31.19	56.97
Our Multi Agent	30.75	65.90
Our Multi Agent Scene	28.49	61.07
Social GAN [13]	27.25	41.44
Our Individual D	25.81	39.72
Our joint D	23.05	34.65
Desire [20]	19.25	34.05
Sophie [28]	**16.27**	**29.38**

Table 2. Quantitative results on Stanford Drone [25] dataset. Average and Final Displacement Errors are reported.

Figure 5 shows ablative qualitative results. For example, notice that two agents traveling from top are making right turns to the left, which, given their past trajectories, scene contexts and other agents' past, can be inferred. This intention is successfully captured by *Multi Agent Scene*, and *Multi Agent* model also inferred partially from the information of other agents. However, without scene information, the predicted trajectories violate scene constraints: they are traveling in the left of the road instead of in the right. In the

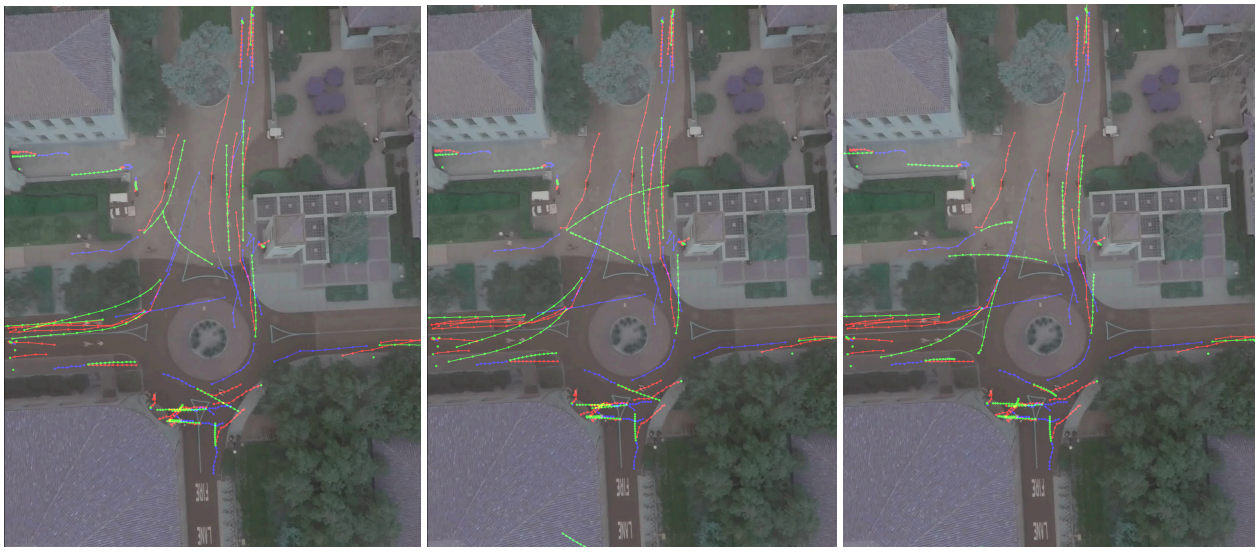

Figure 6. Ablative qualitative results on Stanford Drone [25] dataset. From left to right are results from *Multi Agent Scene*, *Multi Agent*, and *LSTM*. Blue lines show past trajectories, red ground truth, and green predicted. All results come from the qualitative validation dataset.

contrast, although one of the predicted agents *LSTM* makes a right turn following the agent's own past curvature, most of the predicted behavior make little sense.

6. Discussion

We proposed an architecture for trajectory prediction which models scene context constraints and social interaction in a spatial-centric representation, rather than the agent-centric approach more commonly used in the literature. Our motivation was that scene context constraints and social interaction patterns are invariant to the absolute coordinates where they take place; these patterns only depend on the relative positions among agents and scenes. Convolutional layers are suited to modeling these kinds of position-invariant spatial interactions by sharing parameters across space, while recent approaches like Social Pooling [1, 13] or Attention mechanisms [31] cannot explicitly reason about spatial relationships among agents and cannot reason about these relationships at multiple spatial scales. Our spatial-centric fusion module with multi-grid convolutional layers models this ideally. Our spatial fusion approach, to our best knowledge, is the first approach which fuses information from a static scene context with multiple dynamic agent states and retains the their spatial structure throughout the reasoning process for trajectory prediction.

We applied our model to two different trajectory prediction tasks to demonstrate its flexibility and capacity to learn different types of behaviors, agents, and scenarios from data. In the vehicle prediction domain, our model achieved state-of-the-art results at long-range prediction of vehicle trajectories in the NGSIM dataset. Our adversarially trained stochastic prediction model performed best relative to the maneuver-based approach of [9], suggesting that the representation of the distribution over maneuvers was necessary – whether implicit or explicit. Our ablative studies on our private highway driving dataset showed that representations of both the scene and multiagent interactions were necessary for accurate trajectory prediction for highway driving, which typically involves more complex scene context than NGSIM (greater lane curvature, more entrances and exits, etc.).

Our application to a state-of-the-art pedestrian dataset [25] demonstrated comparable performance with previously published results. Although some recent models achieved greater accuracy than ours [28, 20], all used dramatically different architectures; it is interesting to find that a novel spatial-centric architecture can also achieve a high standard of performance. Future work should examine the factors that influence performance, and the advantages and disadvantages of different architectures.

In future work, we plan to integrate explicit maneuver representations using an auxiliary maneuver classification task. We hope that this will close the gap in performance between our deterministic model, and the model in [9], and also increase the interpretability of our predictions.

Social trajectory prediction is a complex task, which depends on the ability to extract structure from the history of agents' joint motions. Our goal here has been to explore the contribution of spatial-centric representations to trajectory prediction. Though our ultimate goal is accurate trajectory predictions, our hope is that the process of engineering better models will continue to yield further insights into the structure of human interaction.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 2, 4, 5, 7, 8

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*. 2015. 2

[3] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr. A combined model and learning based framework for interaction-aware maneuver prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2016. 2

[4] M. Bansal, A. Krizhevsky, and A. S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *CoRR*, abs/1812.03079, 2018. 2

[5] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. *Computer VisionECCV*, 2012. 2

[6] W. Choi and S. Savarese. Understanding collective activities of people from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1242–1257, 2014. 2

[7] J. Colyar and J. Halkias. Us highway dataset. Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030. 2, 4, 5, 6

[8] N. Deo, A. Rangesh, and M. M. Trivedi. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *arXiv:1801.06523*, 2018. 2, 4, 5, 6

[9] N. Deo and M. M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *IEEE Computer Vision and Pattern Recognition Workshop on Joint Detection, Tracking, and Prediction in the Wild*, 2018. 2, 4, 5, 6, 8

[10] N. Deo and M. M. Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2, 6

[11] K. Diederik and W. Max. Auto-encoding variational bayes. In *ICLR*. 2014. 2, 5

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680. 2014. 2, 4

[13] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4, 5, 7, 8

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 3

[15] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2, 4, 5, 7

[16] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, pages 4565–4573. Curran Associates, Inc., 2016. 2

[17] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the International Conference on Robotics and Automation (ICRA) 2018*, pages 5308–5317, 2016. 2

[18] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. 2

[19] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer. Imitating driver behavior with generative adversarial networks. *Intelligent Vehicles Symposium (IV)*, 2017. 2, 4, 5, 6

[20] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2165–2174, 2017. 2, 5, 7, 8

[21] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *IEEE 12th International Conference on Computer Vision (ICCV)*, 2009. 2

[22] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. *IEEE Transactions on CVPR*, 2009. 2

[23] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arxiv:1411.1784*, 2014. 4

[24] N. Rhinehart, P. Vernaza, and K. Kitani. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision (ECCV 2018), Part of the Lecture Notes in Computer Science book series (LNCS, volume 11217)*, pages 794 – 811, October 2018. 2

[25] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. *European Conference on Computer Vision (ECCV)*, 2016. 2, 4, 5, 7, 8

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 3

[27] K. S. S. Pellegrini, A. Ess and L. V. Gool. Youll never walk alone: Modeling social behavior for multi-target tracking. *IEEE 12th International Conference on Computer Vision (ICCV)*, 2009. 2

[28] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. *arXiv, CoRR*, abs/1806.01482, 2018. 2, 5, 7, 8

[29] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. Car-net: Clairvoyant attentive recurrent network. *arXiv:1711.10061*, 2017. 2

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. 2

[31] A. Vemula, K. Muelling, and J. Oh. Social attention: Modeling attention in human crowds. In *Proceedings of the International Conference on Robotics and Automation (ICRA) 2018*, May 2018. 2, 8

[32] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? *IEEE Transactions on CVPR*, 2011. 2