2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Multi-Head Attention for Multi-Modal Joint Vehicle Motion Forecasting

Jean Mercat[1,2], Thomas Gilles[1,3], Nicole El Zoghby[1],
Guillaume Sandou[2], Dominique Beauvois[2], and Guillermo Pita Gil[1]

***Abstract*— This paper presents a novel vehicle motion forecasting method based on multi-head attention. It produces joint forecasts for all vehicles on a road scene as sequences of multi-modal probability density functions of their positions. Its architecture uses multi-head attention to account for interactions between all vehicles, and long short-term memory layers for encoding and forecasting. It relies solely on vehicle position tracks, does not need maneuver definitions, and does not rasterize the scene as a spatial grid. This allows it to be more versatile than similar model while combining many forecasting capabilities, namely joint forecast with interactions, uncertainty estimation, and multi-modality. The resulting prediction likelihood outperforms state-of-the-art models on the same dataset.**

I. INTRODUCTION

Automation of driving tasks aims for safety and comfort improvements. For that purpose, Autonomous Driving (AD) systems rely on the anticipation of the traffic scene movements. Consequently, motion forecasting is used in AD algorithms such as path planning and target selection. The main obstacle in this task is the human driver behavior that can neither be modeled nor predicted perfectly. It is especially challenging in negotiating situations with many participants where drivers' interactivity plays a determinant role. A technical challenge is to find a representation of the road scene that allows forecasting algorithms to account for interactions within a variable number of observed vehicles. It should do so with an unevenly distributed observation accuracy on a wide partially occluded surrounding area. Occlusions and uneven accuracy do not affect the closely related topic of pedestrian motion forecasting where, in most applications, the observations are not embedded in the scene. Two other aspects are specific to vehicles and should be considered: the importance of the road network structure and of the reaction time due to inertia. This requires the understanding of the road network structure and longer time and distance anticipation.

The unknown driver's decisions and the perception inaccuracies make forecasting uncertainties unavoidable. In that context, another objective is to control the uncertainties of the motion forecasts. The scene uncertainties are characterized with a probability density function that presents modes with dispersion. Modes are local maxima of the Probability Density Function (PDF) of future positions. They stand for occurrences of choices, for example, a driver chooses a lane, or the perception system chooses a classification. Dispersions around each mode represent the continuous uncertainties. They are small errors made at each step of the process such as perception, estimation, and some model approximations. When considering multiple modes, there is a challenging trade-off to find between anticipating a wide diversity of modes and focusing on realistic ones. To meet the need for anticipation in AD systems, forecasting algorithms must control the two kinds of uncertainties within an interaction aware framework.

*This work was made in collaboration with L2S and Renault SAS
[1] Department of data fusion, Technocentre Renault, 78280 Guyancourt, France {Jean.Mercat, Thomas.Gilles, Nicole.El-Zoghby, Guillermo.Pita-Gil}@renault.com
[2] Laboratoire des signaux et des systèmes, Centrale-Supélec, 91192 Gif sur Yvette, France {Jean.Mercat, Guillaume.Sandou, Dominique.Beauvois}@centralesupelec.fr
[3] École polytechnique, Route de Saclay, 91128 Palaiseau Thomas.Gilles@polytechnique.edu

II. RELATED WORK

Learned models for trajectory and maneuver forecast are compared in the survey [1]. Since then, recurrent neural networks mostly using the Long Short-Term Memory [2] (LSTM) architecture have become the standard technology for statistical trajectory forecasting. It has been used with the same kind of hand-crafted interaction features than traditional models such as social force [3] but it failed to generalize to complex situations. This was overcome with the social pooling mechanism in [4] by using a spatial grid representation. It places features computed with LSTMs on a coarse *spatial grid* to allow spatially related sequences to share features. The subsequent work [5], used as a baseline in our application, uses convolutional social pooling on a coarse spatial grid. Spatial grids are representation spaces that are able to account for a variable number of input vehicles without ordering. An extension of the convolutional social pooling made in [6] uses non-local multi-head attention over spatial grids to account for long distance interdependencies. Spatial grid representations limit the zone of interest to a predefined fixed size and the spatial relation precision to the grid cell size. To allow direct interactions without spatial grids, in [7], a feature-wise max-pooling of relative positions encoding similar to the PointNet [8] architecture is used. However, the social context treats all the other agents uniformly whereas each agent should interact with a selection of relevant other agents.

Social attention is a mechanism that allows selective interactions within relevant agents. It is used to make the social context more specific. In [9] a different context vector is built for each agent. Other agent features are sorted according to their relative distance to the target agent in a list. This list has a fixed size of $N_{\max}$ agents and is

sensitive to small variations of other agent positions. A soft attention mechanism is used on this list to produce a context feature vector. To avoid the list ordering sensitivity, [10] uses hand crafted relative geometric features to build a set of normalized weights. The context vector is a convex sum of other agent's feature vectors that is invariant to the ordering. These last three solutions are used within a Generative Aderserial Network (GAN) architecture. Generative models such as GANs and Variational Auto-Encoders (VAE) are able to describe complex distributions. However, they are only able to generate an output distribution with sampling and do not express a PDF. More complex interactions than simple distances and angles should be produced in the context of vehicle forecasting to account for specific behaviors such as following or yielding. This is made using a graph representation of vehicles neighbors in [11]. However it only accounts for local interactions among vehicles within a hand-defined distance threshold. A dot product attention mechanism is produced in [12]. It is inspired from the attention mechanism first developed in [13] for sentence translation. This mechanism allows joint forecast of every vehicle in the observed scenes without spatial limitations. It accounts for long range interactions within a varying number of vehicles and does not require the ordering of the vehicle tracks it takes as input. In [12], this dot product attention is used within a spatio-temporal graph representation of the scene developed in [14]. This representation combines spatial and temporal dependencies that rely on positions, pedestrian relative positions, and time step movements as features. Each are embedded with LSTMs before using the dot product attention for social interactions. This identifies important relations between neighbors to be considered for interactions and combines the pedestrian feature representation with the feature representation of the relations. In this work we use the multi-head extension of this attention mechanism, also from [13], with a different road scene representation. We do not rely on spatio-temporal graphs but on a simple temporal embedding followed by social interactions that allows interactions between more complex feature representations than only temporal and relative dynamics. We show in the application (section VII) that different heads specialize to different and interpretable interaction patterns. Our network outputs a mixture of bivariate Gaussian laws that is more adequate to describe the expected distribution than the simple bivariate Gaussian law from [12] and we show that it produces diversified multi-modal forecasts.

Multi-modal forecasts are expressed as predictions with local probability maxima. Mixture Density Networks defined in [15] are used in [5] that predefines driving maneuvers as prediction modes. Each maneuver mode is matched with one Gaussian component of the mixture and a conditional predictor is trained along with a maneuver classifier. As shown in [16], the various modes in the trajectory data are very complex and numerous. Thus, capturing them with a few predefined maneuvers is not enough. Using Gaussian mixture does not necessarily produce diversified modes. A solution to obtain distinct predicted modes without predefining them is proposed in [17]. However, it changes the optimized objective that no longer maximizes the forecast likelihood. In [18] another solution that preserves the forecast statistics while producing diversified predictions is proposed. However, both methods rely on VAEs that generate prediction samples but not the PDF. In [19], a Multiple-Trajectory Prediction (MTP) loss is used to produce multimodal trajectory predictions without the need for sampling. However, as in [17], this modifies the objective function and alters the forecast likelihood.

To the best of our knowledge, our added contributions are:

- The use of multi-head attention for motion forecasting leading to specialized interactions.
- The combination of long range attention with joint and multi-modal forecasts.
- The unsupervised obtention of diversified multi-modal predictions by directly maximizing the forecast likelihood leading to improved likelihood of the results.

III. Inputs and outputs

The inputs are sequences of all vehicle (x, y) positions in a road scene. At each time t_0, we consider an observation history with a fixed observation frequency and a fixed number of observations n_{hist}. The past trajectory is written $\{(x, y)_k\}_{k=-n_{\text{hist}}+1,0}$. The coordinate system is centered on the ego vehicle position at t_0.

The outputs are n_{pred} sequences of Gaussians mixtures for each vehicle. They are expressed with sextuplets $(\hat{x}, \hat{y}, \sigma_x, \sigma_y, \rho, p)$ for each vehicle, at each forecast step and for each mixture component. It defines a Gaussian component $(\mathcal{N}((\hat{x}, \hat{y}), \Sigma), p)$ with

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}$$

the covariance matrix, and p the mixture weight such that for n_{mix} components, $\sum_{m=1}^{n_{mix}} p_m = 1$.

The forecasting model is a set of functions $pred_\theta : inputs \rightarrow outputs$. The inputs and outputs sets are defined with the cartesian products:

$$\text{inputs} \in \left(\mathbb{R}^2\right)^{n_{\text{hist}} \times n_{\text{veh}}}$$

$$\text{outputs} \in \left(\left(\underbrace{\mathbb{R}^2}_{\hat{x},\hat{y}} \times \underbrace{\mathbb{R}^2_+}_{\sigma_x,\sigma_y} \times \underbrace{[-1,1]}_{\rho} \right)^{n_{\text{mix}}} \times \underbrace{\Delta^{n_{\text{mix}}}}_{p} \right)^{n_{\text{pred}} \times n_{\text{veh}}}$$

$\Delta^{n_{\text{mix}}}$ is the n_{mix} element simplex:

$$\Delta^{n_{\text{mix}}} = \left\{ (p_1, \ldots, p_{n_{\text{mix}}}) \in \mathbb{R}^{n_{\text{mix}}} \;\middle|\; \sum_{i=1}^{n_{\text{mix}}} p_i = 1,\ p_i \geq 0 \ \forall i \right\}$$

The $pred_\theta$ function set is defined as a neural network with weights θ. $pred_\theta$ is equi-variant with permutations along the vehicle axis and it is defined for any numbers of vehicles n_{veh} and forecast steps n_{pred}.

IV. Model architecture

This model uses an encoder-decoder structure. It is based on LSTM networks for encoding and forecasting. We propose to add two multi-head self-attention layers to this architecture to account for interactions. The first attention layer is added after encoding to incorporate current time interactions. The second attention layer is added after forecasting time unrolling. This allows the forecast position sequences to remain coherent with each other.

A. Global architecture

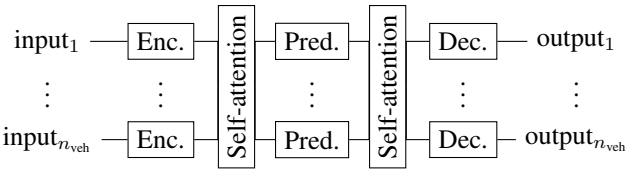


Fig. 1: Block representation of our forecasting model. Inputs are the sequence of past observations of each vehicle. Outputs are the Gaussian mixture forecasts.

The figure 1 breaks the model in four parts: Encoder, Self-Attention, Predictor, and Decoder. The two self-attention layers have similar architectures with different weights whereas the encoder, predictor, and decoder use shared weights.

B. Encoder

The encoder acts as a current state estimation for each vehicle using the past observation sequences. This state is an intermediary vector of the neural network and is difficult to interpret. However, since it should encode the current state with at least the information of position, kinematic state, and interaction features, it should have a sufficient dimension, we chose 120. The input (x, y) position sequences are fed to a one dimensional convolutional layer with a kernel of size 3 sliding over the time dimension that creates sequences of 120 features for each vehicle. This first layer increases the number of features in the vector used for the following computations. A convolution allows this first layer to compute derivatives, smoothed values and other features extracted from successive positions. Then each feature sequence is encoded with a Long Short-Term Memory (LSTM) [2] into a vector of 120 features for each vehicle.

C. Self-attention

The multi-head self-attention layers allow vehicle interactions while keeping independence from their number and ordering. This mechanism is described in [13] where it is applied on sentence translation. In this section we explain its use for vehicle interactions. The computations made by each attention head is represented on figure 2.

Each vehicle should pay attention to specific features from a selection of the other vehicles. This is made with four steps: pulling a subset of specific features, identifying these feature collections, enquiring among identifiers, and gathering the results. Each head produces a different selection of features using a linear projection of the input tensor resulting in the value tensor V. To identify these features, a key tensor K

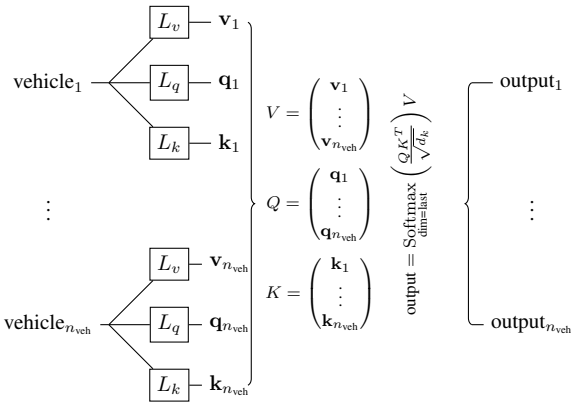


Fig. 2: Schematic representation of one attention head computations. Blocks L_q, L_v, L_k are matrix multiplications of the input vectors.

is associated to each value. Then, each vehicle must select which other vehicle to pay attention to. For that purpose, a query Q is produced to find a selection of keys. The match score between a key and a query is their dot product, it is scaled with the square root of the key dimension $\sqrt{d_k}$ and normalized with a softmax. This produces an attention matrix that contains coefficients close to 1 for matching queries and keys and close to 0 otherwise. The attention matrix is square of size n_{veh}, each coefficient (i, j) is the attention coefficient of vehicle i on vehicle j. Finally, this matrix is used to weight the sum of values from V. Thus, the self-attention computation for each head is written:

$$\text{output} = \underbrace{\underset{\text{dim=last}}{\text{Softmax}}\left(\frac{QK^T}{\sqrt{d_k}}\right)}_{\text{attention matrix}} V \tag{1}$$

The outputs from all heads are concatenated and combined with a linear layer. The resulting tensor is then added to the input as in residual networks.

D. Predictor and decoder

Tensors produced with the self-attention layer are repeated n_{pred} times to be fed as time sequences into a second LSTM layer called predictor. This produces intermediary time sequences with some interaction awareness. Within the feature sequences, vehicle interactions may depend on time. Thus, we placed a second multi-head self-attention layer before feeding the output to the decoder.

Feature sequences are decoded with two linear layers shared for each time step and ReLU activations. Finally, a last linear layer produces the mixture of Gaussian coefficients. The output is described in section III. Let o_i be the i^{th} coordinate of the output tensor before the activation function. To constraint it, the following activation function is applied on each coordinate at every time steps:

$$\begin{aligned}
&\{(\hat{x}, \hat{y}, \sigma_x, \sigma_y, \rho, p)\}_{m=1,n_{\text{mix}}} \\
&\quad = \text{activation}(\{o_1, o_2, o_3, o_4, o_5, o_6\}_{m=1,n_{\text{mix}}}) \\
&\quad = \left\{(o_1, o_2, e^{\frac{o_3}{2}}, e^{\frac{o_4}{2}}, \tanh(o_5), \underset{m \in \text{mix}}{\text{Softmax}}(o_6))\right\}_{m=1,n_{\text{mix}}}
\end{aligned}$$

V. Architecture discussion

A. Multi-head self-attention

The general idea of this architecture is to use the good properties of the key-query self-attention layer to account for interactions. This offers flexibility to the model allowing powerful LSTM models to compute the features and predictions with the fixed size inputs it demands while accepting a varying number of interacting vehicles without ordering. This allows the simultaneous forecast of each vehicle in the scene with vehicle to vehicle interactions. We show in the application section VII-B that multi-head attention produces interpretable interactions with heads specializing on different interaction patterns.

This method could be stated as computationally expensive because with the attention over n_{veh} objects, the overall complexity is $\mathcal{O}(n_{\text{pred}} n_{\text{veh}} d(d + n_{\text{veh}})$. However, in our use cases, the number of objects is lower than $n_{veh} = 30$ and most of the time it remains around 10 which is much lower than the feature dimension $d = 120$.

B. Maneuver free multimodal forecast

The n_{mix} (arbitrary choice) mixture components are diversified solely by the loss minimization. The loss is only the negative log-likelihood (NLL) value averaged over time. Thus, minimizing it pushes the predicted modes toward those of the data distribution. What is forecast is not a mixture of trajectory density functions but a sequence of position mixture density functions. There is a dependency between forecasts at time t_k and at time t_{k+1} but no explicit link between the modes at those times. To simplify, we assume that mixture components centers define local maxima of the probability distributions and can be tracked in time by matching similar mixture coefficient values. They are used as forecast trajectories. Even if the human reasoning and some performance indicators use trajectories, only position PDFs at each time steps are needed for the applications such as path planning and safety assessment.

C. Hyperparameters

This model is defined with a few specific hyperparameters that should be tuned: number of encoded features, number of embedding and decoding layers and their activation functions, number of heads in each self-attention layer, number of mixture component in the output distribution and the error covariance clipping value. Other choices have been made and should be questioned such as the data normalization, the use of shortcut connections with or without layer normalization, the use of LSTM layers, the use of two attention layers and some implicit choices that may have been overlooked. Optimizing the hyperparameters with a thorough process could bring some improvements and help understand the model but is not a part of the present study. In this work, only the general concept was prioritized and the hyperparameters were chosen from experience.

The second self-attention layer might not be strictly necessary in our architecture. However, when it is taken out, the training of the model becomes unstable. This will be studied in more details in a future work.

VI. Loss and performance indicators

The model is trained with the Adam optimizer [20] that minimizes the NLL loss. The usual performance indicators for such forecasting models are root mean squared error (RMSE), final displacement error (FDE), and NLL. Only the NLL accounts for the multi-modal aspect of the forecast, other indicators are merely computed with the most probable trajectory. None of the usual performance indicator is able to judge the trade-off between forecast accuracy and diversity of the predicted modes. Thus no indicator is entirely satisfactory and we also consider the Miss Rate (MR).

In the following equations, for the i^{th} sequence at time t_k, we note (x_k^i, y_k^i) the observed positions, $(\hat{x}_k^i, \hat{y}_k^i)$ the most probable forecast positions, and $(\hat{x}_k^{*i}, \hat{y}_k^{*i})$ the forecast positions that produces the minimum FDE. N is the number of sequences in the subset of the database on which the computation is made.

The RMSE computation is made with equation (2) with

$$\text{RMSE}(k) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_k^i - \hat{x}_k^i)^2 + (y_k^i - \hat{y}_k^i)^2} \tag{2}$$

The FDE values are less sensitive to large errors than RMSE. Its computation is made with equations (3).

$$\text{FDE}(k) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_k^i - \hat{x}_k^i)^2 + (y_k^i - \hat{y}_k^i)^2} \tag{3}$$

The Miss Rate is the rate with which all proposed forecasts miss the final position by more than 2m.

$$\text{MR}(k) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\sqrt{(x_k^i - \hat{x}_k^{*i})^2 + (y_k^i - \hat{y}_k^{*i})^2} > 2} \tag{4}$$

The miss rate is lowered with the addition of relevant components and gives an indication about the trade-off between accuracy and diversity. The 2m threshold is not met if maneuvers such as lane changes are missed by all modes.

The NLL computation, at each forecast time t_k, for each Gaussian component centered on $(\hat{x}, \hat{y})$, with the forecast error $\mathbf{d} = (d_x, d_y) = (x - \hat{x}, y - \hat{y})$ and the forecast error covariance defined with $(\sigma_x, \sigma_y, \rho)$ is written:

$$\begin{aligned}\text{NLL}(d_x, d_y, \Sigma) = &\frac{1}{2}\underbrace{\frac{1}{(1-\rho^2)}\left(\frac{d_x^2}{\sigma_x^2} + \frac{d_y^2}{\sigma_y^2} - 2\rho\frac{d_x d_y}{\sigma_x \sigma_y}\right)}_{\mathbf{d_k}^T \Sigma_k^{-1} \mathbf{d_k}} \\ &+ \ln\left(2\pi \underbrace{\sigma_x \sigma_y \sqrt{1-\rho^2}}_{\sqrt{|\Sigma_k|}}\right)\end{aligned} \tag{5}$$

The time index k is dropped to improve readability. The computation of the overall NLL value for all mixture components is written:

$$\text{NLL}(d_x, d_y, \Sigma, p) = -\ln\left(\sum_{m=1}^{n_{\text{mix}}} p_m e^{-\text{NLL}(d_{x_m}, d_{y_m}, \Sigma_m)}\right) \tag{6}$$

TABLE I: Comparison of MNLL, RMSE, FDE and MR results with baselines using the same dataset. *CSP(M) results were recomputed with some minor modifications for a fair comparison.

Time horizon		1s	2s	3s	4s	5s
MNLL	CV [22]	0.82	2.32	3.23	3.91	4.46
	CSP(M) [5]*	*-0.41*	1.07	1.93	2.55	3.08
	SAMMP	-0.36	*0.70*	*1.51*	*2.13*	*2.64*
RMSE	CV [22]	0.76	1.82	3.17	4.80	6.70
	CSP(M) [5]*	0.59	1.27	2.13	3.22	4.64
	GRIP [11]	*0.37*	*0.86*	*1.45*	*2.21*	*3.16*
	SAMMP	0.51	1.13	1.88	2.81	3.98
FDE	CV [22]	0.46	1.24	2.27	3.53	4.99
	*CSP(M) [5]**	0.39	0.91	1.55	2.36	3.39
	SAMMP	0.31	0.78	1.35	2.04	2.90
MR	CV [22]	0.02	0.20	0.44	0.61	0.71
	*CSP(M) [5]**	0.004	0.03	0.12	0.28	0.44
	SAMMP	0.002	0.02	0.08	0.15	0.23

The mean NLL (MNLL) is the average of the NLL from equation (6) over the test set. Minimizing the NLL loss maximizes the likelihood of the data for the forecast. However, it tends to overfit part of its output. In [21], NLL overfitting has degraded the results, making the NLL value unreliable as a performance indicator. To avoid this problem, we clip the standard deviations with a 10cm minimum value.

VII. Application

This model was implemented using the Pytorch library. The NGSIM datasets US-101 and I-80 and its pre-processing were taken from the published code accompanying the article [5]. This also defines the dataset splitting into training, validation, and test sets. Thus, a fair comparison with these results is made. The dataset contains the tracks of all vehicle position on a road segment observed from a camera. The preprocessing produces data that simulates observations from a given vehicle. Each vehicle is alternatively chosen as the observing vehicle. Its neighbors in adjacent lanes and within a 60m road segment are recorded to produce a local road scene centered on the observing vehicle. This road scene is tracked to produce 8 seconds sequences with all positions being recorded at a 5Hz frequency. The 3 first seconds are used as past observations and the 5 next seconds are used as forecast supervision. A typical training time for our model is 20 to 30 hours.

A. Performance indicators comparison

Table I reports results using the performance indicators defined in section VI. All the compared models except for GRIP [11] were trained and computed on the same dataset and evaluated with the same functions. Since CSP(M) only forecasts the observing vehicle trajectory, only the errors for this vehicle are being compared.

Baselines:

Constant velocity (CV): We used a constant velocity Kalman filter with optimized parameters for forecasting on the same data as described in [22].

Convolutional Social Pooling (CSP(M)): We retrained the model from [5]. It uses a maneuver classifier trained with preprocessed data that conditions a predictor for multimodal forecasts. A forecast of the center vehicle trajectory is made with information from its social environment using the convolutional social pooling mechanism. In [5], the model CSP with unimodal forecast gives better RMSE results than the multi-modal forecast CSP(M).

Graph-based Interaction-aware Trajectory Prediction (GRIP): We took the published results from [11]. It uses a spatial and temporal graph representation of the scene to make a maximum likelihood trajectory prediction simultaneously for all vehicles in the scene. It produces the best RMSE results but it does not account for forecasting error covariance estimation nor multimodality.

Social Attention Multi-Modal Prediction (SAMMP): The model described in this article. We chose six mixture components to match the CSP(M) model for a fair comparison.

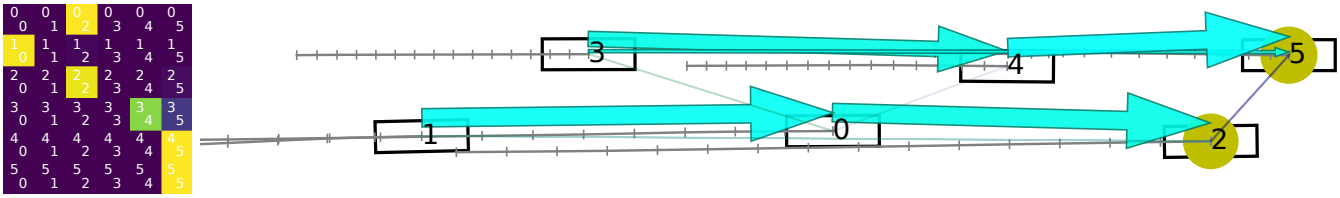
The results from table I show that our model produces a better forecast likelihood than the other models. The lower miss rate shows that the mixture components learned with our model are more relevant than the maneuver definition made in the CSP(M) model and that the forecasts are well diversified. Our RMSE results does not improve upon the results from the GRIP model when we consider the most probable trajectory. However, the most probable trajectory is not always the closest one from the ground truth. The RMSE of the best matching trajectory among the six output trajectories from our model (found using the ground truth) are much lower (respectively 0.31 0.71 1.20 1.80 2.55). Thus, the comparison with the results from GRIP are unclear because its lower RMSE could be caused by mode averaging on their part or by an imperfect mixture probability coefficient evaluation on our part. This could be answered by comparing the miss rates from GRIP with the miss rate of the most probable trajectory from our model.

B. Attention interpretation

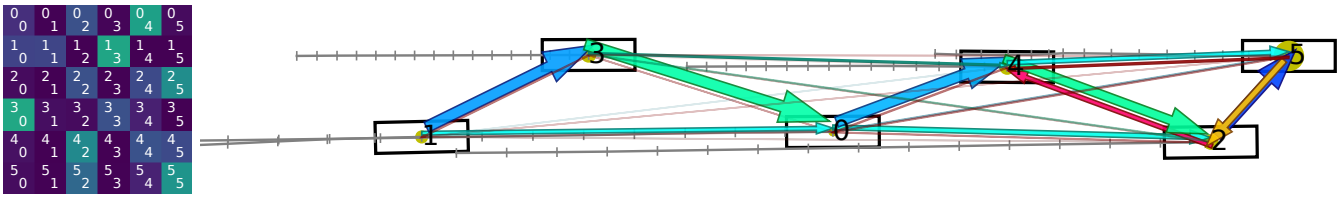
The attention matrices give insights about the importance of some interactions. Some of the head roles can be rationalized by looking at the attention matrix it produces in different contexts. For example, after most tested trainings of the model, one of the heads is strongly specialized in front vehicle attention such as the one in figure 3a. The main attention link always goes from one vehicle to the vehicle in front of it, or to itself if there is no front vehicle. In a few cases, no head is specialized in front vehicle attention, however, one head is specialized in rear vehicle attention. Most of the experiments produced another head matching more or less strongly the closest front vehicle in any lane, such as the one on figure 3b. Other heads also specialize but the interpretation is less clear because the attention is spread over many vehicles. There is often a distinction between front and rear attention and a distinction of lanes.

C. Multi-modal forecasting

On the figure 4, the vehicle 0 aggressively overtakes the vehicle 3. In this situation, the future holds various possibilities. The overtake could be aborted or be made less aggressively, also the last observations of acceleration



(a) Head specialized in front vehicle attention



(b) Head mainly specialized in closest vehicle attention

Fig. 3: A driving scene top view representation with all observed vehicles with their past positions in gray and the attention matrix for two heads of the first attention layer. The attention that vehicle i is giving to j is drawn as an arrow from i to j, and a circle when $i = j$ with widths proportional to the attention coefficient and a color varying with the arrow angle. Attention is also visible as color from purple to yellow in the i, j coefficient of the matrices on the left.

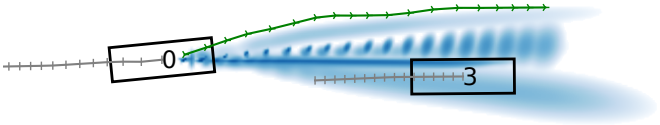


Fig. 4: Another driving scene top view representation. Superposed forecasts are represented in blue shades in log scale. Actual future positions are represented with a green line.

and turning could be the results of perception errors. This example has been chosen because it shows multiple lateral modes that are easier to visualize than the more common longitudinal modes. The NLL loss training is enough to produce a multimodal output matching those possibilities. Using an unmodified NLL loss prevents biases in the forecast distribution that a different loss function may cause and it indeed leads to lower NLL values.

VIII. How to extend this?

The simplest extension is to add observations on each vehicle such as velocity, orientation, size or blinkers. Another extension is to match various object classes such as cars and trucks with specific encoders, predictors, and decoders to allow inter-class interactions. These adaptations can easily be made because our forecasting algorithm is model-free.

Our application and the one from [5] both work with NGSIM US-101 and I-80 datasets. They are composed of highway straight roads observed from above with a camera. This simplistic configuration does not allow training nor testing of the road network understanding. However, this is an important challenge that our model could consider. In the present architecture, self-attention produces keys, queries, and values from the same input to transform the input value. It is possible to produce an asymmetric attention mechanism to consider additional inputs, such as lane center lines discretized as a sequences of reference points. The keys and values can be produced for the additional input with equation 7 whereas the query, as before, relates to the vehicles.

$$\begin{aligned} \text{encoded}_{\text{ext}} &= \text{extEncoder}(\text{input}_{\text{ext}}) \\ V_{\text{ext}} &= L_{v_{\text{ext}}}(\text{encoded}_{\text{ext}}) \\ K_{\text{ext}} &= L_{k_{\text{ext}}}(\text{encoded}_{\text{ext}}) \\ Q_{\text{veh}} &= L_q(\text{Vehicles}) \end{aligned} \tag{7}$$

The attention is then computed as follow:

$$\text{output} = \underset{\text{dim=last}}{\text{Softmax}}\left(\frac{Q_{\text{veh}} K_{\text{ext}}^T}{\sqrt{d_k}}\right) V_{\text{ext}} \tag{8}$$

The output is a weighted sum of the additional features with specific weights for each vehicle. Adding this to the vehicle feature vectors before the vehicle to vehicle attention layer would allow additional context awareness.

To improve the tracking of trajectories from our output, the trajectories should be defined as the optimal transport path between the Gaussian mixtures in time. A variable finite number of trajectories would be produced as the optimal transport paths that pass through local maxima.

IX. Conclusions

We proposed a road scene forecasting model that produces multimodal probability density functions to jointly forecast all vehicles trajectories of the road scene. Our method generates interpretable social attention coefficients that will be extended to other road scene observations. Results from our approach have outperformed state-of-the-art results with the NLL indicator. This shows a good forecasting capacity as well as a good uncertainty evaluation leading to a preferred trade-off between accuracy and prediction diversity. Future work will include attention of vehicles to lanes and be based on the recently published Argoverse [23] dataset in urban situations. We expect the urban conditions to cover complex cases and highly interactive scenes that are better suited to show the interactive capacity of the proposed solution.

Acknowledgements: We are grateful to Edouard Leurent and to the anonymous reviewers for their contribution.

REFERENCES

[1] S. Lefèvre, D. Vasquez, and C. Laugier, A survey on motion prediction and risk assessment for intelligent vehicles, published in ROBOMECH journal,vol. 1, no. 1, pp. 1-14, 2014.

[2] S. Hochreiter, and J. Schmidhuber, Long short-term memory, Neural computation 9(8), 1997, pp. 1735-1780

[3] D. Helbing, P. Molnar, Social Force Model for Pedestrian Dynamics, Physical review. E, 1998.

[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social LSTM: Human Trajectory Prediction in Crowded Spaces, CVPR 2016.

[5] N. Deo and M. M. Trivedi, Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs, published in IEEE Intelligent Vehicles Symposium, 2018.

[6] K. Massaoud, I. Yahiaoui, A. Verroust-blondet, and F. Nashashibi, Non-local Social Pooling for Vehicle Trajectory Prediction, in proceedings of the IEEE Intelligent Vehicles Symposium (IV), Jun 2019.

[7] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, A. Alahi, Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks, CVPR 2018.

[8] C. R. Qi, H. Su, K. Mo, L. J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017.

[9] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, S. H. Rezatofighi, and S. Savarese, SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints, ArXiv:1806.01482v2, 2018.

[10] J. Amirian, J-B. Hayet, and J. Pettré, Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs, CVPR 2019.

[11] X. Li, X. Ying, M. C. Chuah, GRIP: Graph-based Interaction-aware Trajectory Prediction, ArXiv:1907.07792v1, 2019.

[12] A. Vemula, K. Muelling, and J. Oh, Social Attention: Modeling Attention in Human Crowds, IEEE International Conference on Robotics and Automation (ICRA), pp.4601–4607, 2018.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 2017

[14] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, Structural-RNN: Deep Learning on Spatio-Temporal Graphs, CVPR 2016.

[15] C. M. Bishop, Mixture Density Networks, Neural Computing Research Group Report: NCRG/94/0041, 1994.

[16] O. Shouno, Deep learning,driving style,personalization,topological map,unsupervised learning, IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2018.

[17] Y. Yuan, K. Kitani, Diverse Trajectory Forecasting with Determinantal Point Processes, arXiv:1907.04967, 2019

[18] A. Bhattacharyya, B. Schiele, and M. Fritz, Accurate and Diverse Sampling of Sequences based on a "Best of Many" Sample Objective, CVPR 2018.

[19] H. Cui, V. Radosavljevic, F-C. Chou, T-H. Lin, T. Nguyen, T-K. Huang, J. Schneider, and N. Djuric, Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks, ICRA 2019.

[20] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (ICLR), 2015.

[21] D. Lenz, F. Diehl, M. Truong Le, and A. Knoll, Deep Neural Networks for Markovian Interactive Scene Prediction in Highway Scenarios, presented in IEEE Intelligent Vehicles Symposium, pp. 685-692, Jun. 2017.

[22] J. Mercat, D. Beauvois, N. El Zoghby, G. Sandou, and G. Pita Gil, Inertial single vehicle trajectory prediction baselines with NGSIM dataset, ArXiv, 2019

[23] M. F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, J. Hays, Argoverse: 3D Tracking and Forecasting With Rich Maps, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.