MAP-FORMER: Multi-Agent-Pair Gaussian Joint Prediction

Marlon Steiner[1], Marvin Klemp[1], and Christoph Stiller[1]

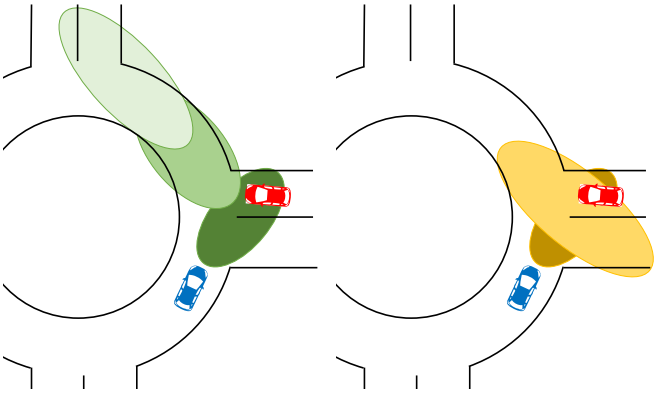
***Abstract*— There is a gap in risk assessment of trajectories between the trajectory information coming from a traffic motion prediction module and what is actually needed. Closing this gap necessitates advancements in prediction beyond current practices. Existing prediction models yield joint predictions of agents' future trajectories with uncertainty weights or marginal Gaussian probability density functions (PDFs) for single agents. Although, these methods achieve high accurate trajectory predictions, they only provide little or no information about the dependencies of interacting agents. Since traffic is a process of highly interdependent agents, whose actions directly influence their mutual behavior, the existing methods are not sufficient to reliably assess the risk of future trajectories. This paper addresses that gap by introducing a novel approach to motion prediction, focusing on predicting agent-pair covariance matrices in a "scene-centric" manner, which can then be used to model Gaussian joint PDFs for all agent-pairs in a scene. We propose a model capable of predicting those agent-pair covariance matrices, leveraging an enhanced awareness of interactions. Utilizing the prediction results of our model, this work forms the foundation for comprehensive risk assessment with statistically based methods for analyzing agents' relations by their joint PDFs.**

I. INTRODUCTION

Motion planning has a strong dependency to motion prediction. Hence, every planned trajectory is influenced by a guess of how the current scene will evolve in the next seconds. The prediction of future trajectories is a non-trivial task: Every action an agent takes, influences its neighboring and scene-related agents, and thus propagates information through the scene. Accordingly, there is an interdependency between actions of all agents in a scene. For the risk assessment of trajectories it is therefore crucial to represent the statistical dependencies between agents in their predicted trajectories. We determine these dependencies by developing a model, which is able to predict agent-pair covariance matrices for the x and y coordinates of both vehicles (Fig. 1).

This work is focused on the prediction fundamentals, required for a statistical risk assessment method of trajectories. The motion prediction task can be distinguished between motion prediction (MP), conditional motion prediction (CMP) and goal-conditioned motion prediction (GCP). [1] provides a very descriptive figure to portray the differences of the three motion prediction tasks. A similar illustration is provided in Fig. 2. The figure shows the motion prediction tasks as three grids of white and blue pixels with the time axis on the horizontal and the agent axis on the vertical. While the white pixels symbolize that the corresponding value can be accessed by the model, the blue pixels are hidden to the model and must be predicted. In the classic MP, the past trajectories of all agents are known by the model and the future must be predicted. This can be done jointly respectively in a "scene-centric" manner or marginal for only one agent. The former one is currently experiencing increasing interest, especially because of its higher significance compared to just predicting marginal trajectories.

In CMP, the whole future of the ego-agent is known, and all other agents must be predicted. While CMP can be a valid assumption for marginal trajectory planning of an ego-agent, we implement MP, since the goal is to predict scene-centric and not focusing on a single agent. Also, this method might



(a) Mode 1: green (b) Mode n: yellow

Fig. 1: Representation of the prediction model's output: One Gaussian joint PDF for every time step and every mode (1a, 1b) of an agent-pair based on the predicted covariance matrices. The different shadings of the ellipses (joint PDFs) represent the consecutive time steps. Due to visualization reasons, 2d ellipses are used instead of 4d Gaussian PDFs as the model actually predicts. Combining the modes with uncertainty weights results in a Gaussian mixture PDF.

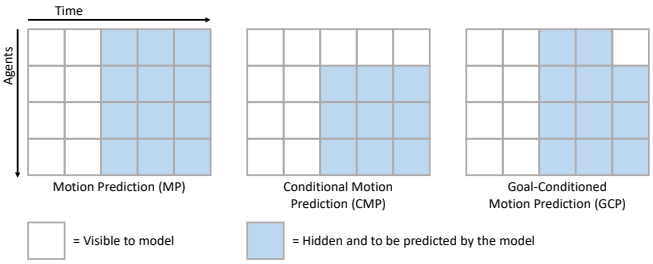


Fig. 2: Different tasks in motion prediction. Here the first two columns represent the past and the last three the future.

[1]Marlon Steiner, Marvin Klemp and Christoph Stiller are with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany {marlon.steiner,marvin.klemp,stiller}@kit.edu

Accepted for publication in Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Jeju Island - Korea, 2-5 June 2024.

be deprecated, since this does not reflect the reality where driving requires multidirectional responses.

Finally, GCP also defines one agent as the ego-agent but in contrast to CMP it only provides information about its last step. We think, that the assumption made in GCP is only valid for a comparatively long prediction horizon, otherwise the provided goal position is less an information of destination but more an information of how the scene is evolving. E.g, considering a prediction horizon of $5\,\mathrm{s}$ in an interactive scene would implicitly provide the prediction module the information of how the interaction is solved. This gives the model more than just routing information, which are in reality not accessible and thus limiting the responsiveness of real traffic.

We choose the rounD dataset [2] because roundabouts represent highly interactive traffic scenarios, which is desirable for our purpose. The dataset contains tracks of $13\,746$ agents (including cars, vans, trucks, buses, pedestrians, bicycles, motorcycles) and has a total length of six hours. Further, the rounD dataset provides high definition (HD) maps with nodes and edges representing properties of the road. We have also deliberately avoided the usage of standard datasets like WAYMO OPEN MOTION DATASET [3], ARGOVERSE [4] or NUSCENES [5] because we explicitly do not focus on the best prediction results regarding a leaderboard but provide a social aware method to predict agent-pair covariance matrices.

To the best of our knowledge we are the first predicting covariance matrices for agent-pairs. Compared to just predicting these for single agents, this provides further statistical information about the relations of agents. To summarize, the main contributions of this work are:

1) A novel model architecture of combining spatial, interaction and temporal information with a GNN and a factorized Transformer for motion prediction.
2) Predicting covariance matrices for agent-pairs in a scene-centric manner, so that multivariate Gaussian joint PDFs can be constructed for all agent-pairs in a scene.
3) A covariance matrix formulation, that can be used in machine leaning, while guarantying its mathematical properties and a corresponding multivariate Gaussian negative log likelihood loss formulation.
4) Proposal of using predicted covariance matrices as a foundation for statistical interactivity and risk analysis.

II. RELATED WORK

The huge amount of publications in the field of motion prediction reflects its enormous relevance in research and application. While our long-term goal is the usage of prediction models in motion planning and especially in risk assessment of trajectories, we still provide a quick overview about recent research in MP.

Marginal Prediction: MP started with predicting marginal future trajectories, i.e. trajectories for single agents in a scene. Different approaches have been used to build the prediction models: Most recent approaches use graph-based methods with GNNs, like VECTORNET [6], LANEGCN [7] and LANERCNN [8] and MULTIPATH++ [9] or TRANSFORMER-based [10] architectures, like WAYFORMER [11]. While [6], [7] and [8] state results on marginal prediction metrics, the authors claim, that joint prediction is possible with their models. Earlier approaches made predictions based on CNNs, e.g. MULTIPATH [12] and [13]. All those prediction models consider multi-modality and therefore output multiple modes of trajectories.

Joint Prediction: In joint prediction, Transformer-based models like SCENE TRANSFORMER [1] and AGENTFORMER [14] dominate over other architectures regarding the prediction metrics. Those models predict future trajectories in a scene-centric manner, so that they output multi-modal futures for all agents in a scene. An example for GNN-based joint prediction is the JFP model [15]. Instead of outputting a joint prediction as a single distribution like in [1], agents get modeled pairwise in [15] and based on that, the whole joint distribution is build up sequentially.

Due to its superior performance, Transformer-based architectures gain increasing interest and have been widely and successfully used for MP tasks. Nevertheless, there is still ongoing research in GNN-based architectures due to their ability of modeling the environmental and social components.

The latest research direction regarding MP was proposed by [16]. They make use of language models (LMs) for predicting joint future trajectories with their model MOTIONLM. Therefore, multi-agent rollouts over discrete motion tokens are leveraged, capturing the joint distribution over multimodal futures.

Gaussian Prediction: Next to predicting just coordinate-based trajectories in a marginal or joint manner, research also focuses on predicting PDFs (mostly Gaussians). This section provides literature, which is most related to our work. Examples for predicting Gaussian mixture (GM) PDFs are WIMP [17], SAMMP [18], CBP [19] and WAYFORMER [11]. Like most of the approaches in MP, they also perform multi-modal predictions. Those models share the fact that they predict Gaussian PDFs for single agents and therefore, provide uncertainty information only for a respective agent. Both, the WIMP model [17] and the CBP model [19] are GNN-based approaches and perform multi-modal prediction as a CMP task with social, environmental and temporal information. The SAMMP model [18] is solely based on vehicle position tracks, and utilizes multi-head attention mechanisms in combination with LSTMs for its prediction.

Prediction in Roundabout Scenarios: As we evaluate our method on roundabouts in the rounD dataset, this section lists papers which also perform prediction on roundabouts. [20], [21] and [22] also train their prediction models on the rounD dataset. Another dataset providing roundabout scenarios is the INTERACTION dataset [23]. Examples for prediction models evaluated on this dataset are [24] and [25]. Except of [25], all presented papers train and compete their models on marginal prediction.

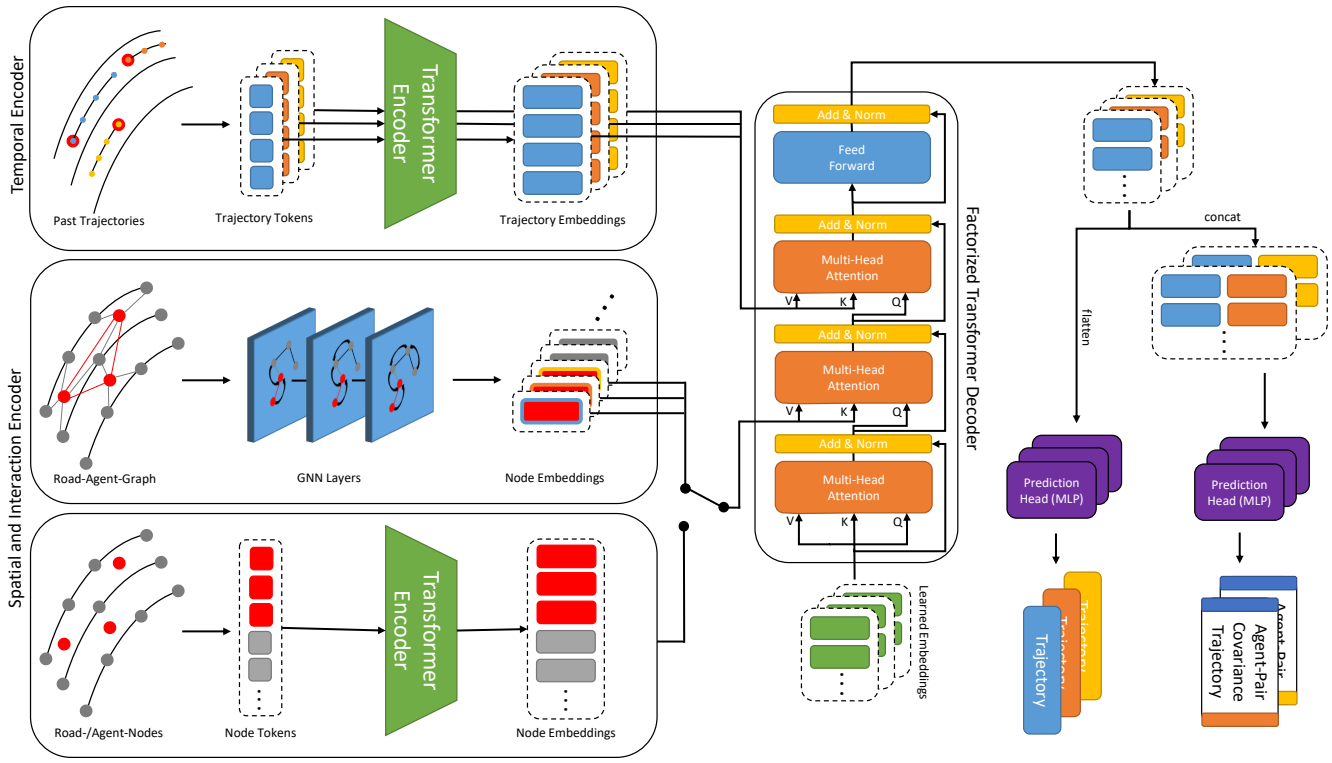


Fig. 3: Network architecture of our motion prediction model. We use a *TEnc* (top left) and the different models can switch between either a GNN-based *SaIEnc* (middle left), a Transformer-based *SaIEnc* (bottom left) or no *SaIEnc*. The red points in the encoders represent the agents. The colors blue, orange and yellow associated with the tokens, embeddings and trajectories represent the corresponding agents.

III. Method

In comparison to the presented Gaussian prediction models, which model multiple modes of each agent as a GM PDF, we are predicting multi-modal PDFs for all relevant agent-pairs, simultaneously. This makes our approach more suitable for assessing risk of trajectories, because the agent-pair PDFs provide more relational information about an agent-pairing. Also, the ego-agent could, e.g. analyze PDFs of other agent-pairs for its own motion planning. Additionally, our approach differs from [17] and [19] that we do scene-centric joint prediction instead of marginal conditional prediction.

Our multi-agent-pair Transformer model (MAP-Former) is composed of four main modules (see Fig. 3): (1) The *Temporal Encoder* uses a Transformer encoder to embed the past trajectories into high dimensional space. (2) The second module is the *Spatial and Interaction Encoder*, which uses a graph as input and extracts its information with either a GNN-based architecture or a Transformer-based architecture. (3) The *Factorized Transformer Decoder* applies cross-attention to the outputs of (1) and (2) with learned embeddings for the future trajectories. (4) The last module produces the predictions and consists of two sets of n linear prediction heads: The first set predicts multiple modes of future trajectories for every agent in a scene. While the second set predicts the parameters of a covariance matrix for every agent-pair corresponding to their trajectories. Both is predicted simultaneously in a single feed-forward pass. The agent-pair covariance matrix prediction is the core of our work, since it allows us to model Gaussian joint PDFs for all agent-pairs in a scene.

A. Temporal Encoder (TEnc)

The *TEnc* processes information about the past trajectories of all agents (red points in Fig. 3) in a scene. For every agent A, the corresponding past trajectory is visualized in a different color, which matches to the color of the embeddings in the rest of the figure. Taking every agent's past trajectory point as a separate token and stacking the tokens of different agents into different vectors, produces the input to the encoder. As the encoder, a Transformer encoder is used. To process the tokens in the encoder, every token gets converted into a unique embedding, so that the past trajectory embedding is of the shape $[A, n_{\text{pastTimeSteps}}, d_{\text{model}}]$. Here d_{model} represents the embedding dimension used over all modules in the model. The encoder applies self-attention to each embedding within a trajectory and again outputs embeddings for every agent and every time step.

B. Spatial and Interaction Encoder (SaIEnc)

To include structural and relational information of a scene, we use a second encoder which either is a GNN-based

SaIEnc or a Transformer-based *SaIEnc*.

***GNN-based SaIEnc**:* Since, GNNs have proven strong performance (e.g. [6], [7]) in learning structural relations, we implement a GNN-based *SaIEnc* to capture further contextual information. For this encoder, we provide a directed road-agent-graph consisting of nodes and edges as input. Nodes represent agents (red points in Fig. 3) and also structural elements of the road (gray points). Edges serve as a connection of contextually coherent nodes, so that connected nodes can aggregate information from each other. The road-graphs are build up from HD maps.

The agent-graph is implemented as a fully connected graph (see Fig. 3), where every agent can aggregate information from all other agents. To enable the aggregation of map information, every agent is further connected to all road-graph nodes within a radius of $r = 5\,\mathrm{m}$ to its center. All nodes and edges are defined by an individual feature vector, characterizing their properties or in case of an edge, the type of connection.

The GNN layers are implemented based on the GIN [26] architecture, where the feature vector $h_v^{(k)}$ of node v in layer k is calculated as follows:

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}\right).$$

The sum is formed over the set $\mathcal{N}(v)$ of neighboring nodes u of the node of interest v. It can be guaranteed that different nodes with different neighborhoods are always mapped as different embeddings, since the MLP and the sum operator, which operates on multisets, are injective. A GNN layer can be seen as a message passing algorithm, where every node receives messages from its neighbors and aggregates them into a new feature vector. For every GNN layer k added to the network, the receptive field consequently increases by one "hop".

The original GIN implementation is not meant to process edge features. Therefore, we use the PyTorch Geometric [27] version of the GIN network (GINE), which can also handle edge features in the message passing.

The output of the GNN are embeddings for every node in the graph of shape $[n_{\mathrm{nodes}}, d_{\mathrm{model}}]$, with encoded information about their k-hop neighbors. Fig. 3 visualizes the agent embeddings bordered with the same colors as in the *TEnc*. The node embeddings of the road-graph (gray points) are discarded.

***Transformer-based SaIEnc**:* As an alternative to the previous encoder, we use a Transformer-based encoder to capture the contextual information of the scene [28]. Here, the nodes of the road-agent-graph are directly used as input. The Transformer encoder applies self-attention between the node embeddings, to extract the structural and relational information between the nodes.

C. Factorized Transformer Decoder

The *Factorized Transformer Decoder* aggregates the information of both, the *TEnc* and the *SaIEnc*, and outputs the final embeddings per agent. We use learned embeddings (green in Fig. 3) of the shape $[A, n_{\mathrm{futureTimeSteps}}, d_{\mathrm{model}}]$ as another input to the decoder, which attach to the encoder embeddings.

First, self-attention is applied among the learned embeddings to identify the relevant relations between the time step embeddings per agent. The output of the self-attention is then used as the query in the first cross-attention block. Here, the node embeddings from the *SaIEnc* serve as agent specific values and keys. Thus, the learned embeddings can attend to the relevant structural and social information. In the next step, the output is again used as the query for the second cross-attention block. The values and keys are taken from the *TEnc*, so that the learned embeddings get updated by attending to the past trajectory embeddings. Analogous to [10] the whole decoder block is repeated N times.

D. Multihead Agent-Pair Prediction

This module takes the generated embeddings of the *Factorized Transformer Decoder* as its input. For the trajectory prediction this information is directly fed into $n = 6$ MLP heads to predict multiple modes of future trajectories. Whereas, for predicting the agent-pair covariance matrices, we first form agent-pairs. Therefore, we choose one agent per scene serving as the "ego-vehicle", whose embedding is then concatenated with all other agent embeddings. E.g. in Fig. 3, the blue embedding (blue agent) is concatenated with the orange embedding respectively the yellow embedding. The concatenation results in an embedding of shape $[A-1, n_{\mathrm{futureTimeSteps}}, 2 \cdot d_{\mathrm{model}}]$. Our architecture also allows to extend the concatenation for all existing agent-pairs in a scene. In the next step, the agent-pair embeddings are fed into n MLP prediction heads, resulting in agent-pair covariance matrices Σ corresponding to the predicted trajectories.

To guaranty, that the predicted covariance matrices fulfill the requirements on symmetry and positive-definiteness, we utilize the properties of the Cholesky decomposition: A symmetric and positive-definite matrix can be decomposed with a lower triangle matrix $\mathbf{L}$ and a positive-definite diagonal matrix $\mathbf{D}$ to:

$$\Sigma = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}.$$

Adapted to our case with four coordinates as random variables, the matrices $\mathbf{L}$ and $\mathbf{D}$ can be constructed as following:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & c & 1 & 0 \\ d & e & f & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \hat{\sigma}_{x_1}^2 & 0 & 0 & 0 \\ 0 & \hat{\sigma}_{y_1}^2 & 0 & 0 \\ 0 & 0 & \hat{\sigma}_{x_2}^2 & 0 \\ 0 & 0 & 0 & \hat{\sigma}_{y_2}^2 \end{bmatrix}.$$

With this formulation of the covariance matrix Σ, we need to predict ten parameters: $\hat{\sigma}_{x_1}\, \hat{\sigma}_{y_1}\, \hat{\sigma}_{x_2}\, \hat{\sigma}_{y_2} \in \mathbb{R}^+$ and the parameters a, b, c, d, e, $f \in \mathbb{R}$ of the matrix $\mathbf{L}$. The standard deviations σ of the covariance matrix cannot directly be chosen by the $\hat{\sigma}$ in $\mathbf{D}$, but these parameters significantly influence the standard deviations. Standard deviations take on values $\sigma > 0$ per definition. This property is modeled

by using a *Softplus* activation function. Afterwards, we shift the output by adding a fixed bias, under which the standard deviation is physically not reasonable.

E. Multivariate Gaussian Negative Log Likelihood Loss

By taking the predicted coordinates μ and the covariance matrix Σ of a single time step and considering the four coordinates x_1, y_1, x_2, y_2 as random variables X, the density function of a multivariate Gaussian PDF can be constructed:

$$f_{\mathbf{X}}(x_1, y_1, x_2, y_2) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu)\right)}{\sqrt{(2\pi)^k \det(\Sigma)}}.$$

Here k represents the dimension of X. As a modification of the Gaussian negative log likelihood loss (GNLL), we form the multivariate case of this loss (MGNNL):

$$\begin{aligned} loss &= -\log\left(f_{\mathbf{X}}\right) \\ &= \log\left(\sqrt{(2\pi)^k \det(\Sigma)}\right) + \left(\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu)\right). \end{aligned}$$

The GNLL loss is derived from the assumption, that the target values are normally distributed around the predicted values. The loss measures how well the predicted PDF explains the predicted values. It encourages the model to not only make accurate predictions but also provide appropriate uncertainty estimates and thus fit a reasonable distribution.

This loss formulation is the core of our work, since the goal is to receive a covariance matrix for all future time steps of every agent-pair. Due to the properties of the Cholesky decomposition, this matrix is always symmetric and positive-definite. Thus, the inversion of the covariance matrix Σ^{-1} for the loss calculation can be guaranteed.

IV. Evaluation

As mentioned before, we train our models on the rounD dataset. Since we perform agent-pair prediction, we skip all frames in which only a single agent occurs. The maximum number of agents recorded in the dataset for a single frame is 25. Hence, our model predicts 2-25 agents jointly, depending on the frame. We predict trajectory points in a frequency of $5\,\mathrm{Hz}$ and provide $1\,\mathrm{s}$ of history to the model.

For the evaluation of our prediction results, we use the standard metrics but in an extended way, so that they are capable of capturing the scene-centric joint prediction:

- Minimum Scene-Centric Average Displacement Error (minSADE) in meters: The ADE is calculated as the average Euclidean L2 distance between all points of a predicted trajectory and the corresponding points of the ground truth. The SADE is therefore the mean over all ADEs of a specific mode. Therefore, the min refers to the mode that provides the minimum SADE.
- Minimum Scene-Centric Final Displacement Error (minSFDE) in meters: The FDE is the L2 distance between the endpoint of a predicted trajectory and its ground truth. Analogous to the SADE, the SFDE is the mean over all FDEs of a specific mode. Consequently, the minSFDE takes the minimum over all predicted modes.
- Scene-Centric Miss Rate (SMR): A prediction counts as a "miss", when the FDE is larger than $2\,\mathrm{m}$. In our case, the SMR is calculated as the number of scenarios where the FDE of at least one agent of the mode with the lowest SFDE is larger than $2\,\mathrm{m}$, divided by the total number of scenarios.

Table I shows the performance of different models on the rounD dataset. We propose three different models resulting from the MAP-FORMER architecture (Fig. 3): First, the MAP-FORMER (Baseline), which only uses the *TEnc* as an encoder. Second, the MAP-GRAPHFORMER, which uses the *TEnc* and the GNN-based *SaIEnc*. And third, the MAP-FORMER (full), which uses the *TEnc* and the Transformer-based *SaIEnc*.

For comparison, we implement a simple joint prediction CNN Baseline [29]. To the best of our knowledge, we are the first performing joint prediction on the rounD dataset. Therefore, the results of three marginal prediction models ([20], [21], [22]) are provided (gray background in Table I).

Adding information about the scene structure to our model, as done with the *SaIEnc*, improves the prediction performance. While the MAP-GRAPHFORMER represents an improvement over the MAP-FORMER (Baseline), the MAP-FORMER (full) outperforms the given joint prediction models in all metrics. The results of the marginal models and the joint models are not directly comparable. Anyway, the MAP-FORMER (full) competes with the best marginal prediction model (N-ODE2 [21]) in the longer prediction horizon. Only in the short prediction horizon, the MAP-FORMER (full) is outperformed by the SSP-ASP [20] model. Notably, the SSP-ASP and the N-ODE2 models get a history of $3\,\mathrm{s}$ instead of $1\,\mathrm{s}$ like our models.

In the following, we shortly present the results for the agent-pair covariance matrix prediction. A covariance matrix (4×4) can be seen as a 2×2 block matrix with four 2×2 blocks. The blocks on the main diagonal represent the marginal covariance matrices of the predicted trajectory points of the respective agents within an agent-pair. While the block on the upper diagonal (identical to the lower diagonal block) represents the covariance matrix between the agents. For a first proof of concept, we sum up the absolute values of the upper diagonal blocks of the predicted agent-pairs' covariance matrices. These sums are shown in Fig. 4 as lines with variable thickness connecting the agents. Where the thickness represents a measure of the dependency respectively the interactivity between agent-pairs. The predictions originate from our MAP-FORMER (full) model.

In this example we build up agent-pairs based on agent 0 and visualized possibly interesting dependencies in Fig. 4. Agent 6 inside the roundabout is a human, maintaining the planted area. We can see that agent 0 has the lowest dependency with agent 2, which is reasonable because agent 2 is spatially far away from agent 0 and there are also agents between them. Also, agent 8 has a low dependency with agent 0, since agent 8 has already entered the roundabout and agent 0 is about to exit. Agent 10 and 11 have a higher and quite similar dependency with agent 0. Given that agent

TABLE I: Results on rounD dataset for $n = 6$ prediction heads (or modes) and a prediction horizon of $t = 3\,\mathrm{s}$ and $t = 5\,\mathrm{s}$ with "scene-centric" (S) metrics. The models with gray background are only evaluated on marginal prediction (not scene-centric). The best joint prediction results are highlighted in bold and the best marginal prediction results are underlined.

Method	minSADE ↓		minSFDE ↓		SMR ↓	
	3 s	5 s	3 s	5 s	3 s	5 s
SSP-ASP [20]	<u>0.17</u>	1.25 (6 s)	<u>0.74</u>	4.61 (6 s)	-	-
N-ODE2 [21]	-	<u>0.98</u>	-	<u>3.09</u>	-	<u>0.35</u>
Extended DGNN [22]	1.68	-	1.69	-	-	-
CNN Baseline [29]	1.46	3.57	4.30	10.29	1.00	1.00
MAP-FORMER (Baseline) (Ours)	0.71	1.49	1.84	4.03	0.90	0.97
MAP-GRAPHFORMER (Ours)	0.60	1.30	1.59	3.48	0.83	0.96
MAP-FORMER (full) (Ours)	**0.52**	**1.20**	**1.38**	**3.22**	**0.75**	**0.95**

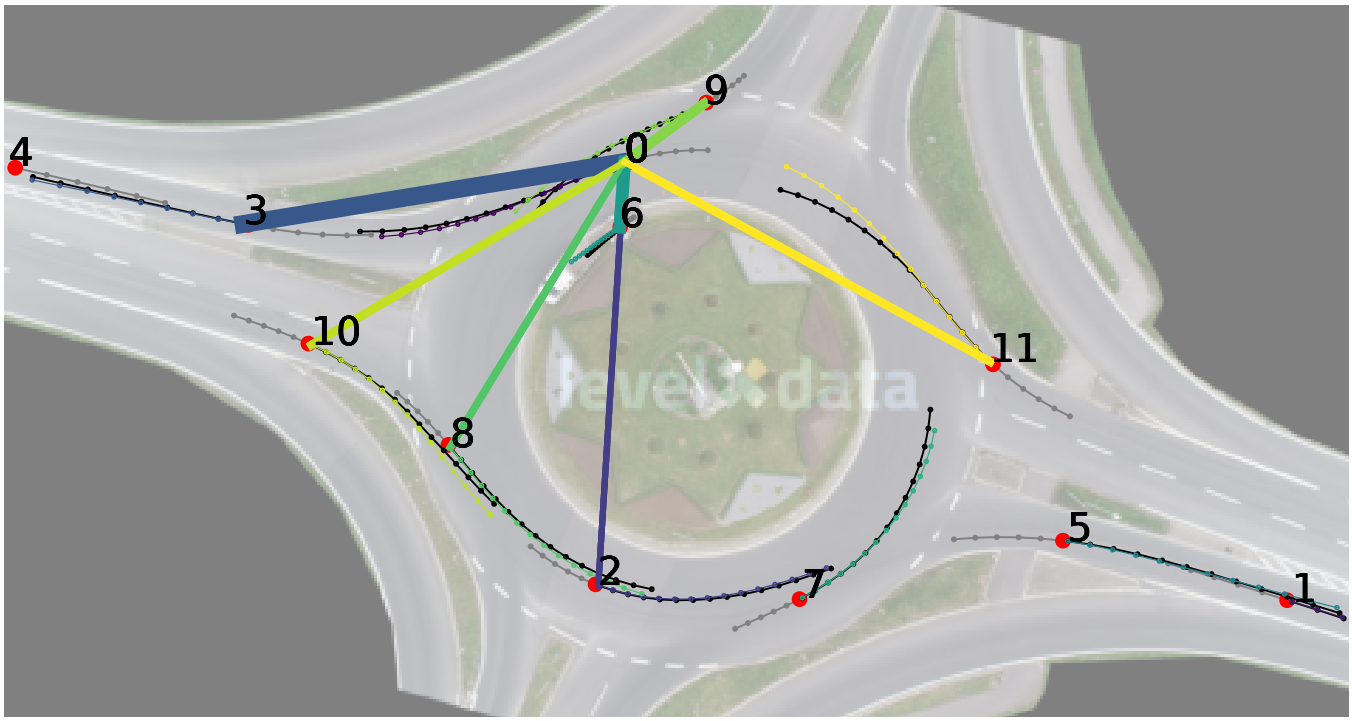


Fig. 4: Visual prediction results: The figure shows a scene from rounD [2] with twelve agents (red points). For every agent the figure provides its past trajectory (gray), its ground truth (black) and its predicted trajectory for $t = 3\,\mathrm{s}$ (colored). The lines, connecting the agent-pairs, represent the upper diagonal blocks of the predicted covariance matrices and therefore describe the dependencies between agent-pairs.

0 is about to exit the roundabout, agent 10 and 11 are in a similar situation to agent 0. The highest dependencies to agent 0 have – in ascending order – agents 6, 9 and 3. Agent 9 is spatially close to agent 0 and therefore, a high dependency is reasonable. It is possibly unsure, if agent 6 intends to cross the road, and it is additionally spatially close related to agent 0, so a high dependency is also reasonable. Agent 3 is spatially further away from agent 0, but it can be considered as the leading vehicle and thus has a high dependency with agent 0. A comprehensive statistical analysis of the agent-pair Gaussian PDFs will be part of a follow-up work, so we will not go into more detail about the agent-pair correlation analysis.

V. CONCLUSIONS AND FUTURE WORK

In this work we presented a novel view on motion prediction in the context of motion planning and risk assessment. We propose to predict statistical information between agent-pairs. Therefore, we developed a multi-agent-pair prediction model, capable of predicting not only coordinates of a trajectory but predicting joint covariance matrices. These can be used for modeling agent-pair Gaussian PDFs and calculate dependencies between them. While Gaussian PDFs for single agents only allow a statement about the uncertainties of each agent's coordinates, our method also provides information about their dependencies. Based on this prediction approach, a follow-up paper will comprehensively analyze agent interactivity and risk utilizing the probabilistic joint

PDFs generated by the predicted covariance matrices.

ACKNOWLEDGEMENTS

This work is accomplished within the project "AUTOtech.agil" (FKZ 01IS22088) and the financial support from the German Federal Ministry of Education and Research (BMBF) is acknowledged.

REFERENCES

[1] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, *et al.*, "Scene Transformer: A unified architecture for predicting future trajectories of multiple agents," in *International Conference on Learning Representations*, 2022.

[2] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The rounD Dataset: A Drone Dataset of Road User Trajectories at Roundabouts in Germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[3] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9710–9719.

[4] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, *et al.*, "Nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[6] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, *et al.*, "VectorNet: Encoding HD Maps and Agent Dynamics From Vectorized Representation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 522–11 530.

[7] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, *et al.*, "Learning Lane Graph Representations for Motion Forecasting," *CoRR*, vol. abs/2007.13732, 2020, arXiv: 2007.13732.

[8] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 532–539.

[9] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 7814–7821.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[11] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 2980–2987.

[12] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[13] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, *et al.*, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 2090–2096.

[14] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9813–9823.

[15] W. Luo, C. Park, A. Cornman, B. Sapp, and D. Anguelov, "Jfp: Joint future prediction with interactive multi-agent modeling for autonomous driving," in *Conference on Robot Learning*, PMLR, 2023, pp. 1457–1467.

[16] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, *et al.*, "Motionlm: Multi-agent motion forecasting as language modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8579–8590.

[17] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *CoRR*, vol. abs/2008.10587, 2020. arXiv: 2008.10587.

[18] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9638–9644.

[19] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying Driver Interactions via Conditional Behavior Prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3473–3479.

[20] F. Janjoš, M. Dolgov, and J. M. Zöllner, "Self-supervised action-space prediction for automated driving," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2021, pp. 200–207.

[21] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, "Evaluation of differentially constrained motion models for graph-based trajectory prediction," *arXiv preprint arXiv:2304.05116*, 2023.

[22] G. Daoud, M. El-Darieby, and K. Elgazzar, "Prediction of autonomous vehicle trajectories in turnaround scenarios," in *2023 10th International Conference on Dependable Systems and Their Applications (DSA)*, IEEE, 2023, pp. 606–613.

[23] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[24] A. Ścibior, V. Lioutas, D. Reda, P. Bateni, and F. Wood, "Imagining the road ahead: Multi-agent trajectory prediction via differentiable simulation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 720–725.

[25] D. Grimm, P. Schörner, M. Dreßler, and J.-M. Zöllner, "Holistic graph-based motion prediction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 2965–2972.

[26] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" *CoRR*, vol. abs/1810.00826, 2018, arXiv: 1810.00826.

[27] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[28] R. Wagner, O. S. Tas, M. Klemp, and C. F. Lopez, *Redmotion: Motion prediction via redundancy reduction*, 2023. arXiv: 2306.10840 [cs.CV].

[29] N. Nikhil and B. Tran Morris, "Convolutional neural network for trajectory prediction," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.