IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

DiPA: Probabilistic Multi-Modal Interactive Prediction for Autonomous Driving

Anthony Knittel[1], Majd Hawasly[1], Stefano V. Albrecht[1,2], John Redford[1], Subramanian Ramamoorthy[1,2]

***Abstract*—Accurate prediction is important for operating an autonomous vehicle in interactive scenarios. Prediction must be fast, to support multiple requests from a planner exploring a range of possible futures. The generated predictions must accurately represent the probabilities of predicted trajectories, while also capturing different modes of behaviour (such as turning left vs continuing straight at a junction). To this end, we present DiPA, an interactive predictor that addresses these challenging requirements. Previous interactive prediction methods use an encoding of k-mode-samples, which under-represents the full distribution. Other methods optimise closest-mode evaluations, which test whether one of the predictions is similar to the ground-truth, but allow additional unlikely predictions to occur, over-representing unlikely predictions. DiPA addresses these limitations by using a Gaussian-Mixture-Model to encode the full distribution, and optimising predictions using both probabilistic and closest-mode measures. These objectives respectively optimise probabilistic accuracy and the ability to capture distinct behaviours, and there is a challenging trade-off between them. We are able to solve both together using a novel training regime. DiPA achieves new state-of-the-art performance on the INTERACTION and NGSIM datasets, and improves over the baseline (MFP) when both closest-mode and probabilistic evaluations are used. This demonstrates effective prediction for supporting a planner on interactive scenarios.**

***Index Terms*—Autonomous Vehicle Navigation, Motion and Path Planning, Deep Learning Methods**

I. Introduction

PREDICTION of the future motion of surrounding road users is essential for the safe operation of an autonomous vehicle (AV). Road scenarios such as intersections, merges and roundabouts require significant interaction between agents in the scene, where agent behaviour is influenced by the presence of nearby agents, as well as reactions to actions that other agents take. In order to support planning, a predictor needs to estimate the future states of the surrounding road users based on observations of their recent history, and to estimate the risk of conflict for possible ego actions.

A planning system used in interactive scenarios needs to consider different possible actions that other vehicles may take, and the futures that result from different actions. In order to explore these futures, a supporting predictor needs to be computationally fast, and to provide accurate predictions that represent the expected distribution of future states of each agent. Many combinations of actions may be possible, so an interactive predictor needs to be fast in order to allow different futures to be explored.

Existing predictors addressing this task have encoded predictions using a fixed number of mode samples, for example using 6 predicted trajectories encoded as center positions [1], [2], [3]. These are evaluated using minimum average- or final-displacement error (minADE/FDE) and miss-rate (MR) (see Section IV-B). These measures compare the closest predicted mode with the ground-truth, and are important for demonstrating that predictions closely capture distinct modes of behaviour observed in the data.

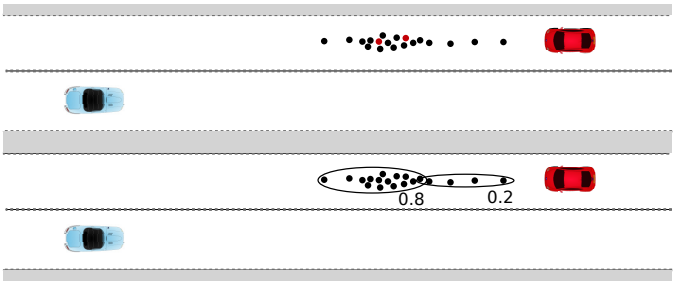

Fig. 1. Top: Use of k-mode samples (red, k=2) under-represents the distribution of future positions (black). This prevents effective planning by underestimating states which are reasonably likely to occur. Bottom: A GMM encoding, with associated mode weights, provides a more accurate representation of the full distribution by covering a wider range of samples.

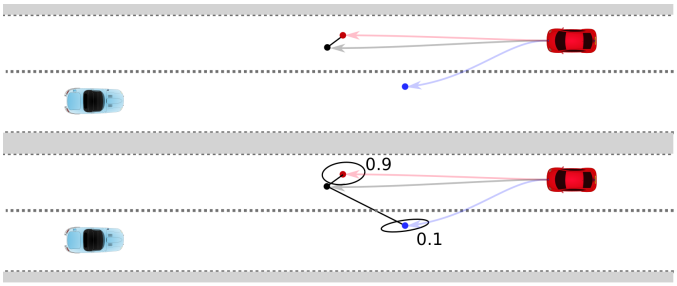

Fig. 2. Top: Optimising for closest-mode evaluations can allow unrealistic predictions to be over-represented. For an instance of data (black dot), the closest predicted mode (red) is evaluated while additional modes (blue) can predict unrealistic behaviours without penalty. Unlikely predicted modes interfere with planning, for example causing an emergency break to avoid a predicted collision that is unlikely. Bottom: Optimising for both closest-mode and probabilistic evaluations penalises unlikely predictions, while minimising over- and under-representation.

Manuscript received: October 8, 2022; Revised March 7, 2023; Accepted May 16, 2023.

This paper was recommended for publication by Ashis Banerjee upon evaluation of the Associate Editor and Reviewers' comments.

[1]Five AI Ltd, UK. contact: anthony.knittel@five.ai
[2]School of Informatics, University of Edinburgh, Edinburgh, UK

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

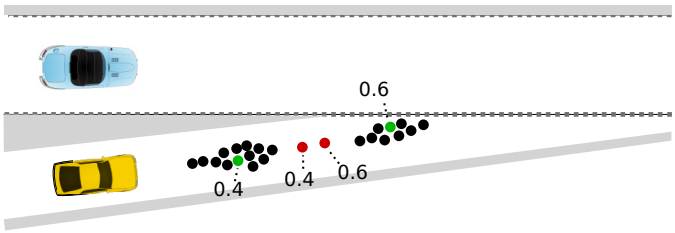

Fig. 3. A merge scenario produces a bi-modal distribution (black samples). Optimising closest-mode (minADE/FDE) evaluations favours diverse predictions (green), while probabilistic (predRMS) evaluations favour predictions close to the mean (red), that minimise the penalty of incorrect mode estimates. Solving both requires diverse predictions with the ability to accurately estimate mode probabilities.

A limitation of this sample-based encoding is that it does not represent the full distribution of expected future positions, and as such many variations are under-represented (Fig. 1). A further limitation is that probabilities of predicted modes are not considered. When training a model based on closest-mode evaluations, additional predicted modes (other than the closest) do not affect scoring, which allows the predictor to predict behaviour modes that are unlikely to occur. Each predicted mode has equal weight, which results in over representation of unlikely predictions (Fig. 2).

These limitations can be addressed using a Gaussian Mixture Model (GMM), which represents the full predicted distribution, along with probability estimates of each mode. This is preferred over increasing the number of samples, as GMMs provide a compact encoding of the distribution and a practical means of evaluating the probability distribution. Previous methods [4], [5] have used GMMs on the NGSIM dataset, which are evaluated using negative-log-likelihood (NLL) evaluations. Further methods have used mode probability estimates [6], [7] which are evaluated using predicted-mode RMS (predRMS) evaluations (see Section IV-B).

Probabilistic and closest-mode evaluations provide complimentary measures that are more informative than either alone, and are analogous to precision and recall in binary classification. We argue that an effective predictor for interactive scenarios needs to optimise both measures, to demonstrate that it is able to closely capture distinct behaviour modes, while also accurately representing probabilities. This is a challenging task as different evaluation measures are supported by contradictory prediction strategies. Closest-mode evaluations (minADE/FDE/MR) favour diverse predictions, while probabilistic evaluations (predRMS, NLL) favour conservative predictions close to the mean of expected behaviours, where the cost of incorrect mode estimates is minimised (Figure 3). Optimising both evaluation approaches together demonstrates accurate multi-modal prediction, and reduces the over-representation of unlikely predictions seen in Figure 2.

To that end, we present DiPA (Diverse and Probabilistically Accurate) – a fast method for predicting in interactive scenarios using a GMM encoding, that is able to optimise both objectives together, by producing a diverse set of predictions with accurate probability estimates. This allows distinct behaviours to be accurately modelled, while producing an accurate representation of the full trajectory distribution. This improves over previous methods [1], [3] using closest-mode evaluations on the INTERACTION dataset [8], and improves over previous methods [7], [4] using probabilistic evaluations on NGSIM [9]. DiPA also improves over a baseline method (Multiple-Futures Prediction (MFP)) [5] when comparing both closest-mode and probabilistic measures together. This demonstrates a predictor that is suitable for supporting an AV planner in interactive scenarios.

Beyond highlighting the importance of evaluating predictors with both closest-mode and probabilistic evaluations, the key contributions are: 1) a fast prediction architecture with a flexible representation that processes agent interactions in wide-ranging road layouts, that produces high accuracy predictions on interactive scenarios, 2) a training regime that supports a diverse set of predicted modes using a GMM-based spatial distribution, with accurate probability estimates, and 3) a revision to the NLL measure for evaluating GMM predictions, to correct for an important limitation.

II. Related work

A number of different structures have been used for prediction of agents in road scenes, including graph-, goal- and regression-based methods.

StarNet [1] represents the scene and agents using vector-based graphs, and uses a combined representation of agents within their own reference frame and from the points of view of other agents. Further graph-based methods such as [10], [3], [11] combine map information and agent positions into a common representation, commonly processed with a Graph Neural Network [12] in an encoder-decoder framework. These methods allow encoding the static layout of the scene and various agents in a generalisable way, and have shown good results on closest-mode prediction.

Goal-based methods [13], [14], [15], [16], [17], [18] identify a number of potential future targets that each agent may head towards, determine likelihoods of each, and produce predicted trajectories towards those goals. Flash [7] uses a combination of Bayesian inverse-planning and mixture-density networks to produce accurate predictions of trajectories in highway driving scenarios. Goal-based methods use the map to inform trajectory generation, and can use kinematically-sound trajectory generators. However, this can lead to limited diversity on other factors such as motion profile and path variations compared to data-driven methods.

Regression-based methods use representations that directly map observations to predicted outputs. SAMMP [4] produces joint predictions of the spatial distribution of vehicles, using a multi-head self-attention function to capture interactions between agents. Multiple-Futures Prediction (MFP) [5] models the joint futures of a number of interacting agents, using learnt latent variables for generating predicted future modes. Mersch et al. [19] present a temporal-convolution method for predicting interacting vehicles in a highway scenario where neighbouring agents are assigned specific roles based on relative positions to a central agent. These regression-based methods can be fast and accurate, but may have limited gen-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

eralisability to different layouts when role-based representation of inputs is used.

Existing interactive prediction using the INTERACTION dataset have demonstrated good results based on closest-mode evaluations (minADE / FDE / MR) [1], [2], [3]. These have typically used a prediction encoding using a fixed number of modes, each represented as a trajectory sample. Optimising closest-mode evaluations produces diverse predictions, which closely capture distinct modes of behaviour.

Methods using the NGSIM dataset have shown good results on probabilistic evaluations (predRMS, NLL) [4], [19], [5], [7]. These have used a range of encodings including k-mode-samples [7], or GMM models [4], [6], [5]. Optimising these measures allows the probability distribution of predictions to be captured, however may not capture distinct behaviour modes closely.

The importance of balancing prediction diversity and probabilistic accuracy has been recognised in [4] which trains a GMM model with NLL loss and shows improved diversity against prior art measured by MR. Rhinehart et al. [20] examined the generation of paths that are both diverse, to cover instances in the dataset, and precise, to minimise inconsistency with the data, using a specific cross-entropy term per objective. This balance is also addressed in generative CVAE models such as [21] which uses a trajectory sampler trained to extract diverse and plausible samples generated by the model.

To address the limitations of closest-mode and mean evaluation measures, [20] propose the use of information-based cross-entropy evaluations. The importance of evaluating with both displacement-error and NLL-based evaluations has also been recognised by [22], for evaluating trajectories produced by a generative model. Measures of diversity and precision have also been explored by [23] using closest-mode and mean mode evaluations, performed on joint predictions of various agents in a scene. A limitation of this approach is the lack of probabilistic weighting of predicted modes.

A useful interactive predictor requires (1) speed, which can be achieved by minimising unnecessary complexity; (2) an accurate encoding of the full probability distribution over trajectory predictions, as provided by a GMM; and (3) accurate predictions that capture distinct behaviour modes with an accurate distribution, as measured by closest-mode and probabilistic evaluations. Addressing these factors together is challenging, as there are trade-offs between solving each, for example increasing diversity to capture distinct behaviours introduces a cost with estimating the probability distribution accurately. We demonstrate that the proposed DiPA method addresses this joint task, in a generalisable way that can be applied to the various scenes of interactive scenarios.

III. Proposed method

DiPA uses an encoder-decoder architecture, where interactions between agents are captured using a Graph Neural Network (GNN). An overview of the network structure is shown in Figure 4. In order to support speed of processing, and to identify the essential elements needed, the design is focused on the minimal complexity that is needed to produce high-quality predictions. Agents are encoded based on observed histories such as positions and velocities, and each of the agents are treated as symmetric entities in an unordered set, with no need to assign specific roles based on relative positions. This allows flexible comparisons between agents to be performed and enables generalisability to widely varying scenarios including roundabouts, junctions, highways and other road topologies with a varying numbers of agents in diverse arrangements. Predictions are performed jointly on up to 20 agents at a time, while for evaluation purposes a single agent is used for each instance, where the surrounding neighbours are provided for context.

Inputs to the model are the observed histories of each agent (positions, orientations and speeds), and agent features including dimensions and type. The model produces predictions as a multi-modal GMM, represented with a 2D Gaussian distribution for each timestep.

Observed states for each agent are encoded using temporal convolution layers, and interactions between agents are processed using an edge-based GNN. Edge features between pairs of interacting agents are produced by broadcasting agent encodings using concatenation, which are processed with MLP layers for each agent $\times$ agent pair.

This design has been chosen to emphasise the ability to directly process relative values between pairs of interacting agents, such as encodings of positions, velocities and orientations, which are trained based on regression. This is in contrast to standard GNN approaches [12], which use an encoding per agent, where interactions between agents are processed using summation (or other reductions) of encoding messages passed from neighbouring nodes. The proposed approach has similarities with the processing of entities in an unordered set used in PointNet [24].

Reduction over edges (agent pairs) for each agent (node) produces a summary encoding for each agent, while reduction over agent nodes produces a scene context encoding, which allows properties of the scene to influence agent predictions. The agent-context representation is decoded to produce predicted trajectory positions, spatial distribution parameters and mode weight estimates. This design captures the important elements of processing agent predictions with interactions, while removing unnecessary complexity.

A. Training

A typical approach for training GMM predictions is to minimise a NLL loss, such as the score used for evaluation in (10). When this loss is used, the spatial distribution parameters are updated using the predicted mode weight. Inaccuracy in predicted weights produces randomness in mode training weights, resulting in mode convergence and loss of diversity.

We propose a novel training method that improves prediction diversity, which allows distinct modes of behaviour to be captured, and produces accurate estimates of probabilities. Training is performed with 1) a spatial distribution loss for training spatial distribution parameters, and 2) a mode weight estimation loss for training predicted mode weights.

Training mode weights (used with both losses) define the extent that each predicted mode will be updated based on an

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

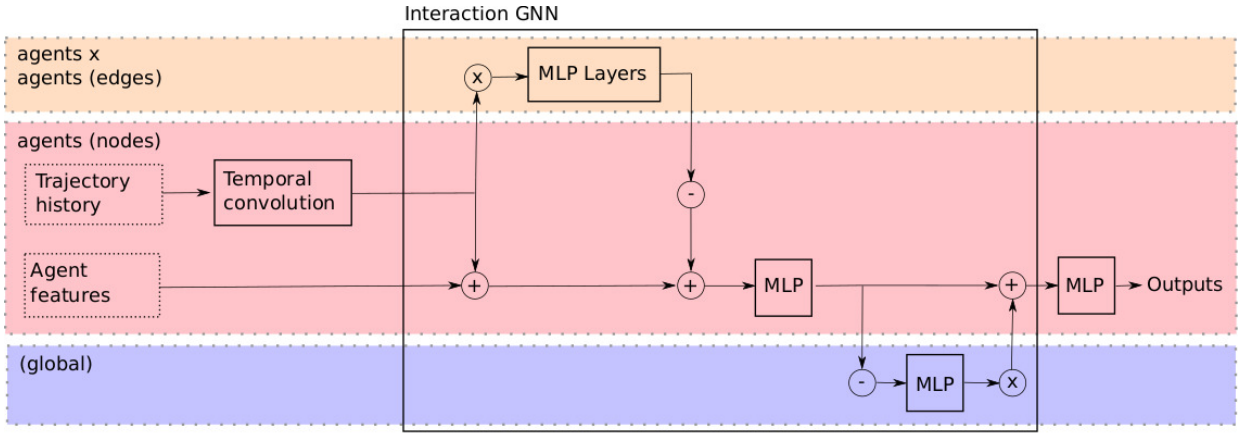

Fig. 4. Network diagram of DiPA model. Trajectory history and agent dimensions are inputs. The following symbols represent $\otimes$=broadcasting, $\oplus$=concatenation, $\ominus$=max reduction. Outputs are predicted trajectories, spatial distribution parameters and mode prediction weights.

observation, where a flat training distribution leads to convergent modes while a biased distribution encourages diversity. Mode weight distributions $W \in \mathbb{R}^M, \sum_m^M W_m = 1$ represent the weighting of each mode for training or prediction. Training mode weights W_r are a combination of the closest mode weight W_c and posterior mode weight W_p, using a proportion weighting $k_r = 0.5$, chosen experimentally as described in Section V-A.

$$W_r = (1 - k_r)W_c + k_r W_p \tag{1}$$

W_c is a strongly biased (one-hot) distribution that encourages training of the single most similar mode to the ground-truth. $\mu_{m,t}$ is the predicted trajectory position for mode m at time t, and x_t is the ground-truth.

$$W_{c,m} = \begin{cases} 1 & \text{if } m = \text{argmin}_m(\frac{1}{T}\sum_t^T ||x_t - \mu_{m,t}||) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

W_p is a weakly biased distribution based on the posterior of the observation under the GMM model, and produces a balance of convergent and divergent mode training that facilitates participation of the different modes. W_p prevents one or a few modes from dominating, and reduces sensitivity to initialisation. $\Sigma_{m,t}$ is the predicted covariance matrix.

$$W_{p,m} = \frac{1}{T}\sum_t^T \frac{\mathcal{N}(x, \mu_{m,t}, \Sigma_{m,t})}{\sum_i^M \mathcal{N}(x, \mu_{i,t}, \Sigma_{i,t})} \tag{3}$$

In contrast, MFP [5] uses a combination of posterior and predicted distribution weights for training the GMM, which has a tendency to produce a single dominant mode.

1) Spatial distribution training: The spatial distribution loss (4) minimises the NLL score of an observation x under the predicted model, weighted by the training mode distribution W_r. This trains the parameters of the normal distribution μ, Σ, while W_r is constant.

$$\mathcal{L}_{spatial} = -\frac{1}{T}\sum_t^T \ln(\sum_m^M W_{r,m}\mathcal{N}(x, \mu_{m,t}, \Sigma_{m,t})) \tag{4}$$

The training weight distribution W_r emphasises training modes similar to the observation, supporting mode diversity, in contrast to standard NLL training based on the predicted mode weight distribution.

2) Mode weight estimation training: Two predicted mode weight terms are produced by the model, W_s and W_n, which are based on similarity of trajectory positions, and low spatial distribution error respectively. Separate terms are used as the ideal mode weights can be inconsistent for different objectives. The trajectory-based mode estimation weight W_s is trained with a MSE-based loss, as shown in (5). (W_s is trained while μ is constant)

$$\mathcal{L}_{MSE} = \sum_m^M W_{s,m}\frac{1}{T}\sum_t^T ||x_t - \mu_{m,t}||^2 \tag{5}$$

The spatial distribution mode weight W_n is trained in order to minimise the NLL score, and to approach the training mode distribution W_r, as shown in (6). (W_n is trained and μ, Σ, W_r are constant)

$$\mathcal{L}_{DIST} = -\frac{1}{T}\sum_t^T \ln(\sum_m^M W_{n,m}\mathcal{N}(x, \mu_{m,t}, \Sigma_{m,t})) + D_{KL}(W_r||W_n) \tag{6}$$

The two mode estimation distributions W_s and W_n are based on different objectives, and favour trajectory- and distribution-based evaluations respectively. In order to produce a single prediction that balances these objectives, a weighted average is returned $W_o = (1 - k_n)W_s + k_n W_n$, using $k_n = 0.9$, which has been chosen experimentally as shown in Section V-A, so that the proposed method out-performs prior methods on all tasks.

IV. Experiments

Experiments are conducted to demonstrate that the proposed DiPA method meets the objectives for supporting an interactive planner, which requires fast processing, the ability to capture distinct modes of behaviour, and to accurately capture the probability distribution for predictions. Experiments are conducted on the INTERACTION [8] dataset to compare existing benchmarks using closest-mode evaluations,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

to demonstrate capturing distinct behaviour modes. Experiments on NGSIM [9] compare against prior methods using probabilistic evaluations, to demonstrate the ability to capture the distribution accurately. As there is a trade-off between optimising closest-mode and probabilistic tasks, experiments are conducted on the joint task using both evaluation approaches on each dataset, which is compared against MFP as a baseline (Section IV-A).

1) INTERACTION dataset: The INTERACTION dataset [8] is divided into instances based on each fully-observed agent in each case window, with a 4 second duration. Prediction is performed using a 1 second observed period and 3 second prediction period.

2) NGSIM dataset: The NGSIM dataset contains trajectory tracks for agents in two scenes (US-101 and I-80). Agents are assigned to train/evaluation splits based on vehicle identifier, as used in [6]. Instances are created based on a central agent, for each fully-observed window of 8 seconds (3 observed, 5 future). For each instance up to 20 neighbouring agents are also observed, while agents that have been assigned to different splits are not used for training.

On both datasets global coordinates are used. Pre-processing centers units on the last observed position of the agent to be predicted, with rotation such that the yaw of the prediction agent is zero (at the last observed timestep).

A. Revised implementation of Multiple-Futures Prediction

MFP [5] is a useful baseline as it is an accurate method based on a GMM, allowing comparison on each of the evaluation measures. A limitation of MFP is that it has been implemented using local lane-based coordinates, which are suitable for highway driving involving mostly parallel lanes. This representation is not directly generalisable to more complex scenarios involving intersections, roundabouts and other non-parallel topology, as are used in INTERACTION.

In order to use global coordinates, for consistency each instance is re-framed to be centered on the last observed position and rotated on the orientation of the central agent. A revised neighbour grid is used to allow MFP to operate on widely varying road topologies. MFP represents neighbours using a 13×3 grid of positions in the central and neighbouring lanes, based on distances from the central agent. A comparable neighbour grid is produced based on the central and neighbouring lane patches corresponding with the central agent. All following and preceding lane patches from the central lane patch(es) represent the central lane, and similarly for the neighbouring lanes. Grid spacing distances for each neighbour agent are found based on the nearest midline path, using the progress distance of the neighbour agent relative to the central agent. This defines a neighbour grid similar to that used in MFP, and implements the *MFP-general* method.

B. Evaluation measures

1) predRMS: the RMS error of the most probable predicted mode is calculated for a number of timesteps, over the instances of the dataset N as shown in (7), where μ_i is the predicted position for the most probable mode $i = \arg\max_m(W_m)$ as used in [6], [5], [7]:

$$predRMS_t = \sqrt{\frac{1}{N}\sum_{n=1}^{N} ||x_{n,t} - \mu_{i,t}||^2} \tag{7}$$

We use the same number of modes as used in corresponding minADE/FDE experiments.

2) minADE: evaluates the closest average Euclidian distance between the predicted trajectory mode and the ground truth over a horizon T, while *minFDE* evaluates the closest final position, as follows.

$$minADE = \min_m(\frac{1}{T}\sum_{t=1}^{T} ||x_t - \mu_{t,m}||) \tag{8}$$

$$minFDE = \min_m(||x_T - \mu_{T,m}||) \tag{9}$$

3) Miss-rate (MR): is defined as the percentage of instances where the minimum spatial error on the final timestep is larger than a given threshold, ie $minFDE > k \in \mathbb{R}$. We use a threshold of $k = 2m$ as used in [13], [4].

4) Negative-log-likelihood (NLL): describes the log-probability of observed instances under a predicted distribution. Previous methods [5], [25], [4] use a GMM representation, although NLL can be compared between different representations. Calculation of the NLL score using a GMM is shown in (10). This is represented using a center position $\mu_m \in \mathbb{R}^2$, covariance matrix $\Sigma_m \in \mathbb{R}^{2\times2}$ and weight $W_m \in \mathbb{R}$ for each predicted mode, where $x \in \mathbb{R}^2$ is the ground-truth position.

$$NLL = \frac{1}{T}\sum_{t=1}^{T} -\ln(\sum_{m=1}^{M} W_m \mathcal{N}(x_t, \mu_{m,t}, \Sigma_{m,t}))$$
$$\mathcal{N}(x, \mu, \Sigma) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{10}$$

NLL as a concept is a dimensionless property, however previous results and the evaluation in (10) represent probability density, without reducing to a dimensionless value. Observed samples are points, which have zero probability in a spatial distribution as a result of being a position with no size. It is possible to produce a probability evaluation using an area instead of a point, however the area to use is not well defined or meaningful for the task. As dimensioned probability density values are used, and the NLL measure is determined from this value, it is important to record the units of the density-based NLL property reported. Previous methods have used inconsistent units, in feet [5] and in meters [4], so to address this problem we present units of measurement with reported results.

Another limitation is that in existing definitions NLL is an unbounded quantity, which allows scores on a small number of instances to greatly influence evaluation over the dataset. This is both a theoretical and a practical problem, as for example a dataset may contain a stationary object, where the center of a predicted GMM can be accurately chosen, and the distribution

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE I
PROBABILISTIC PREDRMS SCORES ON NGSIM

Method	predRMS (by time period) $[m]$ 1s	2s	3s	4s	5s
CV [26]	0.76	1.82	3.17	4.80	6.70
CSP(M) [6], [4]	0.59	1.27	2.13	3.22	4.64
GRIP[a] [27]	0.52	1.22	2.05	3.13	4.47
SAMMP [4]	0.51	1.13	1.88	2.81	3.98
Flash [7]	0.51	1.15	1.84	2.64	3.62
MFP [5]	0.54	1.17	1.87	2.71	3.67
DiPA	**0.46**	**1.04**	**1.70**	**2.47**	**3.43**
Trajectory mode weight	0.46	1.04	1.70	2.45	3.39
Spatial mode weight	0.47	1.08	1.79	2.62	3.64
Standard NLL loss	0.44	1.03	1.66	2.48	3.50
Closest-mode training	0.43	0.99	1.64	2.44	3.43
Posterior training	0.43	0.99	1.65	2.47	3.50

[a]Results are adjusted to correct for scoring with RMS with average over spatial dimension values instead of Euclidean distance RMS.

TABLE II
PROBABILISTIC NLL SCORES ON NGSIM (MODES=5)

Method	NLL (by time period) $[\ln m^{-2}]$ 1s	2s	3s	4s	5s
CV [26][a]	0.82	2.32	3.23	3.91	4.46
CSP(M) [6], [4][a]	-0.41	1.07	1.93	2.55	3.08
SAMMP [4][a]	-0.36	0.70	1.51	2.13	2.64
MFP [5]	-0.64	0.71	1.56	2.21	2.74
DiPA	**-1.22**	**0.20**	**1.23**	**2.01**	**2.61**
Non-thresholded	-2.50	-0.14	1.12	1.98	2.60
Trajectory weight	9810.85	3670.04	63.28	29.92	17.52
Spatial weight	-1.24	0.18	1.21	2.00	2.60
Standard NLL loss	-1.36	0.06	1.11	1.91	2.60
Closest-mode	-1.17	0.51	1.60	2.36	2.86
Posterior training	-1.30	0.11	1.16	1.95	2.60

[a]Previously reported NLL results do not use the thresholded NLL score described in Section IV-B4.

width reduced to an arbitrarily high density, bounded only by numerical limits. When represented with a 64-bit float (with limit 5.5×10^{-309}), this can result in a NLL score of -710 for a single instance.

We suggest that a maximum probability density be applied, as for vehicle prediction there is no practical advantage in distinguishing between very tight bounds. Mercat et al. [4] apply a minimum limit to the standard deviation of $\sigma = 0.1m$, for the purposes of avoiding overfitting. We extend this definition to apply to evaluation, where the probability density is capped for each instance, based on the maximum probability density of a normal distribution with $\sigma = 0.1m$, which gives a minimum NLL score of $-\ln(\frac{1}{2\pi 0.1^2})$ (approx. -2.77). This can be used with any probability distribution, including GMMs and raster-based representations.

V. RESULTS

a) Capturing probability distribution: Comparison using probabilistic evaluations on NGSIM are shown in Tables I and II (ablations are below the double line, as discussed in Section V-A). DiPA improves over previous methods on predRMS evaluations, which involves generating a set of predicted trajectories and accurately predicting the most probable mode. Evaluation of the spatial distribution using NLL shows improved probabilistic accuracy with DiPA over previous methods. These experiments show advantages of DiPA for capturing probabilistic predictions on NGSIM.

b) Capturing distinct behaviours: Comparisons using closest-mode evaluations on INTERACTION is shown in Table III. DiPA shows lower error based on the closest mode than the comparison methods. Comparison of closest-mode Miss-Rate (MR) evaluations on NGSIM are shown in Table IV. DiPA shows improved MR evaluation over methods such as SAMMP [4] that are based on standard NLL training, showing improved ability to produce diverse modes that closely cover individual instances of the dataset. These experiments show that DiPA improves over previous methods for accurately capturing distinct modes of behaviour.

TABLE III
CLOSEST-MODE SCORES ON INTERACTION (MODES=6)

Method	minADE_6	minFDE_6	MR_6
TNT [13]	0.21	0.67	–
ReCoG [10]	0.19	0.66	–
ITRA [2]	0.17	0.49	–
GoHome [3]	–	0.45	–
StarNet [1]	0.16	0.49	–
joint-StarNet [1]	0.13	0.38	–
MFP-general	0.43	1.20	0.19
DiPA	**0.11**	**0.34**	**0.02**
Trajectory mode weight	0.11	0.34	0.02
Spatial mode weight	0.11	0.34	0.02
Standard NLL loss	0.18	0.47	0.03
Closest-mode training	0.11	0.33	0.01
Posterior training	0.11	0.36	0.02

c) Combined task: In order to compare the ability to optimise both closest-mode and probabilistic tasks at the same time, results using multiple evaluation measures are shown for MFP and DiPA in Tables I, II and IV. These show that MFP produces accurate probabilistic predictions as measured with predRMS and NLL, however shows relatively high error on closest-mode evaluations. This suggests limited diversity of predictions, which limits the ability to closely match individual instances. Comparison of multiple evaluations on INTERACTION is shown in Tables III, V and VI. This also shows improved results with DiPA against MFP-general on all evaluation measures, showing advantages of DiPA for optimising both tasks together.

Figure 5 shows selected instances from the INTERACTION dataset and predictions produced by DiPA. This shows DiPA's ability to represent the full predicted distribution using a GMM encoding, providing greater coverage of variations compared to the typical k-mode-samples approach, while also capturing distinct modes of behaviour.

The run-time of the model is $\sim$16ms to predict 20 agents at a time (Python/Tensorflow), using a NVidia 2080Ti GPU. This fast run-time for repeat calls allows multiple predictions to be made as part of inference performed by a planner.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

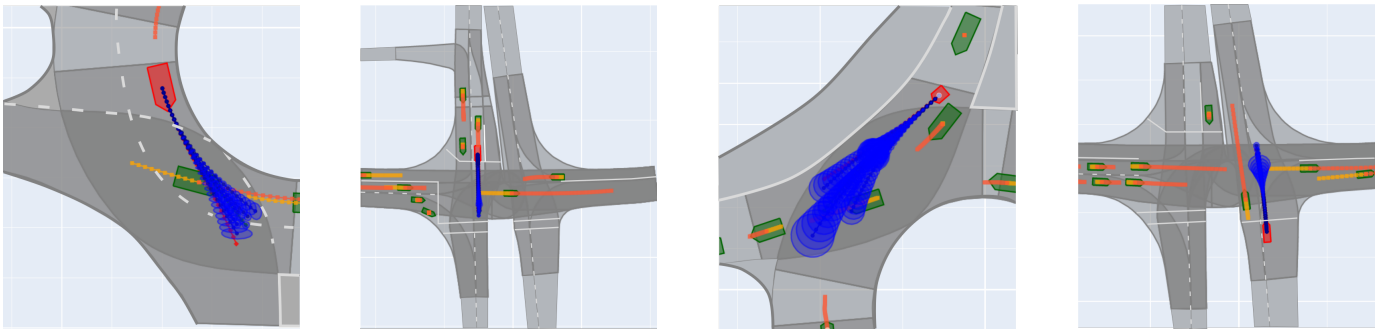

Fig. 5. Qualitative results, showing use of the GMM encoding to represent the full predicted distribution (blue ellipses show $\sigma = 1$). Ego is red vehicle, green vehicles are neighbours, showing history (yellow) and future (orange). L-R: 1. spread to capture variations over chosen paths, 2. narrow spread with variations in speed when crossing intersection, 3. cyclist prediction with large variations in speed and path, 4. distinct modes, with narrow prediction while crossing intersection and also wide spread at slower speeds.

TABLE IV
CLOSEST-MODE SCORES ON NGSIM

Method	minADE_5	minFDE_5	MR_5
CV [26]	_	_	0.71
CSP(M) [6], [4]	_	_	0.44
SAMMP [4]	_	_	0.23
MFP [5]	1.07	2.15	0.40
DiPA	**0.48**	**0.86**	**0.07**
Trajectory mode weight	0.48	0.86	0.07
Spatial mode weight	0.48	0.86	0.07
Standard NLL loss	0.90	1.75	0.32
Closest-mode training	0.46	0.82	0.05
Posterior training	0.51	0.99	0.16

TABLE V
PROBABILISTIC PREDRMS SCORES ON INTERACTION

Method	predRMS $[m]$ 1s	2s	3s
MFP-general	0.21	0.95	2.37
DiPA	**0.11**	**0.47**	**1.28**
Trajectory mode weight	0.11	0.47	1.25
Spatial mode weight	0.12	0.51	1.38
Standard NLL loss	0.11	0.44	1.18
Closest-mode training	0.15	0.50	1.28
Posterior training	0.10	0.46	1.27

TABLE VI
PROBABILISTIC NLL SCORES ON INTERACTION (MODES=6)

Method	NLL $[\ln m^{-2}]$ 1s	2s	3s
MFP-general	-1.87	0.46	2.17
DiPA	**-2.09**	**-0.85**	**0.76**
Non-thresholded	-4.82	-1.67	0.35
Trajectory mode weight	93.20	18.19	12.79
Spatial mode weight	-2.10	-0.87	0.76
Standard NLL loss	-2.27	-0.58	0.95
Closest-mode training	-1.56	-0.87	0.73
Posterior training	-2.22	-0.94	0.70

A. Ablation study

Experiments using variations of DiPA are shown in each result table below the double line. Evaluating with *Non-thresholded* NLL scores show lower error values, particularly for short time horizons that involve narrower error distributions. These low scores can result from tight bounds on a few instances, and thresholded scores are more informative. Predicting with the *trajectory mode weight* W_s alone ($k_n = 0$) favours RMS scores at a cost of NLL evaluations, while the *spatial mode weight* W_n alone ($k_n = 1$) produces lower NLL error with increased RMS errors. The effect of changing the proportion k_n is shown in Figure 6, showing values for the final timestep. A proportion of $k_n = 0.9$ allows effective prediction according to both trajectory- and distribution-based evaluation. *Standard NLL loss* shows a condition where training is performed directly from the NLL loss, using the predicted mode distribution as the training weights. This shows lower error on NLL, but substantially higher error on closest-mode evaluations, showing that it is not able to capture specific behaviour modes as well. *Closest-mode training* based on the mode weight W_c only ($k_r = 0$) shows lower closest-mode errors but higher NLL error on NGSIM, which contains more noise than INTERACTION. During development, we have found that using W_c only can lead to one or a few modes dominating, and is expected to be sensitive to initialisation. *Posterior training* based on W_p ($k_r = 1$) shows lower predRMS and NLL error but higher closest-mode error, showing it is not as effective at capturing distinct behaviour modes. A balanced setting ($k_r = 0.5$) provides a reliable approach that improves over prior methods on all measures.

VI. CONCLUSION

In order to support an AV planner for operating in interactive scenarios, a predictor needs to be fast, to identify distinct modes of behaviour, and accurately represent the probability distribution of predictions. Previous interactive predictors are able to capture distinct modes of behaviour, as measured by closest-mode evaluations, however the k-mode-samples encoding under-represents the full distribution, and misses many variations that can reasonably be expected. In addition, when probability estimates are not reported or used for optimisation, the predictions can over-represent behaviours that are unlikely to occur.

Our proposed DiPA method uses a GMM encoding to represent the full predicted distribution, and uses a novel

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

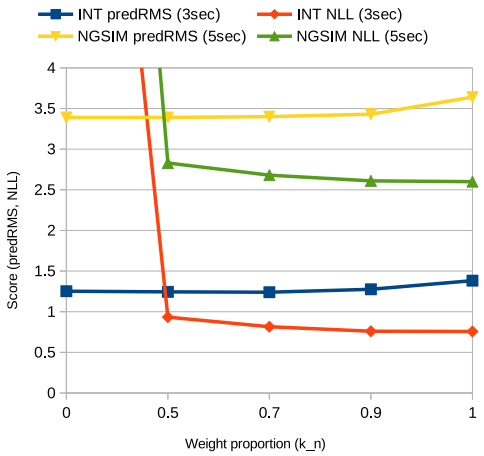

Fig. 6. Effect of changing proportion k_n balancing trajectory- and spatial-distribution-based mode weights.

architecture and training regime that allows learning of distinct modes of behaviour, while also accurately representing the probability distribution. Solving both of these tasks together is more challenging than solving either on its own.

Results on the INTERACTION and NGSIM datasets show DiPA captures distinct behaviour modes and probability estimates better than previous methods. There is a trade-off between these tasks, and comparison using both evaluations shows improvement over the MFP baseline on each measure, demonstrating the ability to accurately model the probability distribution of a diverse set of predicted behaviours.

Limitations of DiPA are that it does not use map information, which prevents following a given road layout, and as a regression-based network, the model can also occasionally produce unrealistic predictions. This is currently mitigated using a wrapper to constrain maximum predicted speeds.

DiPA shows fast run times, and produces an accurate encoding of the full distribution of predictions, that minimises both over- and under-representation of predictions. This provides useful predictions for supporting an AV planner in interactive scenarios.

REFERENCES

[1] F. Janjoš, M. Dolgov, and J. M. Zöllner, "StarNet: Joint action-space prediction with star graphs and implicit global-frame self-attention," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022.

[2] 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.

[3] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "GoHome: Graph-oriented heatmap output for future motion estimation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9107–9114.

[4] 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)*. IEEE, 2020, pp. 9638–9644.

[5] C. Tang and R. R. Salakhutdinov, "Multiple Futures Prediction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[6] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.

[7] M. Antonello, M. Dobre, S. V. Albrecht, J. Redford, and S. Ramamoorthy, "Flash: Fast and light motion prediction for autonomous driving with Bayesian inverse planning and learned motion profiles," in *IEEE International Conference on Intelligent Robots and Systems*, 2022.

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

[9] J. Colyar and J. Halkias, "Next generation simulation (NGSIM) vehicle trajectories and supporting data," 2016. [Online]. Available: http://doi.org/10.21949/1504477

[10] X. Mo, Y. Xing, and C. Lv, "ReCog: A deep learning framework with heterogeneous graph for interaction-aware trajectory prediction," *arXiv preprint arXiv:2012.05032*, 2020.

[11] X. Jia, L. Sun, H. Zhao, M. Tomizuka, and W. Zhan, "Multi-agent trajectory prediction by combining egocentric and allocentric views," in *Conference on Robot Learning*. PMLR, 2022, pp. 1434–1443.

[12] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[13] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "TNT: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021.

[14] J. Gu, C. Sun, and H. Zhao, "DenseTNT: End-to-end trajectory prediction from dense goal sets," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.

[15] J. P. Hanna, A. Rahman, E. Fosong, F. Eiras, M. Dobre, J. Redford, S. Ramamoorthy, and S. V. Albrecht, "Interpretable goal recognition in the presence of occluded factors for autonomous vehicles," in *IEEE/RSJ International Conf. on Intelligent Robots and Systems*, 2021.

[16] S. V. Albrecht, C. Brewitt, J. Wilhelm, B. Gyevnar, F. Eiras, M. Dobre, and S. Ramamoorthy, "Interpretable goal-based prediction and planning for autonomous driving," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[17] C. Brewitt, B. Gyevnar, S. Garcin, and S. V. Albrecht, "GRIT: fast, interpretable, and verifiable goal recognition with learned decision trees for autonomous driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[18] C. Brewitt, M. Tamborski, C. Wang, and S. V. Albrecht, "Verifiable goal recognition for autonomous driving with occlusions," in *IEEE ICRA Workshop on Scalable Autonomous Driving*, 2023.

[19] B. Mersch, T. Höllen, K. Zhao, C. Stachniss, and R. Roscher, "Maneuver-based trajectory prediction for self-driving cars using spatio-temporal convolutional networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021.

[20] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting," in *European Conference on Computer Vision*, 2018, pp. 772–788.

[21] 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.

[22] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*. Springer, 2020, pp. 683–700.

[23] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *European Conference on Computer Vision*. Springer, 2020.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[25] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "PiP: Planning-informed trajectory prediction for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020.

[26] J. Mercat, N. E. Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Kinematic single vehicle trajectory prediction baselines and applications with the NGSIM dataset," *arXiv:1908.11472*, 2019.

[27] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3960–3966.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.