2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

NeSyMoF: A Neuro-Symbolic Model for Motion Forecasting

Achref Doula*, Huijie Yin*, Max Mühlhäuser* and Alejandro Sanchez Guinea*

Abstract— **Recent advancements in deep learning have significantly enhanced the development of efficient models for multi-modal path prediction within urban environments, offering approaches to navigate complex environments accurately. Despite their performance, models grounded in deep learning techniques frequently encounter challenges related to interpretability. This limitation not only hampers their practical application but also complicates the process of diagnosing and rectifying errors within these systems, which is a critical factor for ensuring reliability and safety in real-world deployments. In this paper we propose NeSyMoF, a Neuro-Symbolic model for Motion Forecasting, to address this critical gap by combining the predictive power of deep neural networks with the interpretable logic inherent in symbolic reasoning. Data processing in NeSyMoF involves extracting pertinent features from the agent's environment and channeling them into a neuro-symbolic reasoning module. The neuro-symbolic reasoning module generates first-order logic rules that describe and condition the path prediction process, thereby providing clear explanations and intentions behind the forecasts of the model. We evaluate our model with the Argoverse benchmark for path forecasting, as it includes challenging driving situations, necessary to extensively evaluate our model. The results of our evaluation show that NeSyMoF outperforms state-of-the-art interpretable models for single-mode predictions while providing logic-based explanations for its forecasts, that articulate the reasoning behind predictions, making NeSyMoF more adapted for human-centric applications.**

I. Introduction

Forecasting future trajectories is crucial in a multitude of domains, to enhance the safety and efficiency of systems ranging from autonomous navigation to surveillance. The capability to predict the behavior of road users, for instance, significantly contributes to the safety mechanisms of autonomous vehicles, ensuring their ability to navigate complex environments without endangering human lives [1]. Similarly, surveillance systems benefit from trajectory prediction by optimizing their tracking mechanisms for more effective monitoring and security measures [2], [3]. The evolution of deep learning has been instrumental in this regard, providing a foundation for developing models capable of understanding and predicting the motion of various agents, including pedestrians [4], vehicles [5], [6], and cyclists [7].

The applications of path prediction models in real-world systems underscore the necessity of human trust and feedback in these technologies. For technologies to be embraced on a large scale, especially in safety-critical systems, they must be deemed trustworthy by their human users [8]. Trust in such systems is closely tied to the interpretability of the models that power them. Interpretability ensures that users can understand, trust, and effectively interact with the system. This is crucial not only for user acceptance but also for the practical deployment and troubleshooting of these systems. Efforts to enhance interpretability in path prediction models have seen the development of various techniques, such as the semantic labeling of maneuvers to make path predictions more comprehensible [9], [10], the utilization of interpretable feature maps that offer insights into the decision-making process [11], and the conditioning of predicted paths on explicit agent goals and intentions, which aligns predictions with observable intentions [12], [5].

Among the methodologies aiming to bridge the gap between deep learning performance and human interpretability, neuro-symbolic models has emerged as a compelling approach. Neuro-symbolic models integrate the robust learning capabilities of deep neural networks with the clarity and logical structure of symbolic reasoning. This hybrid approach has demonstrated significant promise in tasks requiring nuanced interpretation of scenes and the generation of interpretable navigation programs, particularly for autonomous vehicles [13], [14]. The integration of symbolic reasoning allows these models to provide explanations in a form that is closer to human reasoning, making them especially suited for applications where understanding the "why" behind a decision or prediction is as important as the outcome itself. Despite the potential of neuro-symbolic modeling, it was not adopted in path prediction, to the best of our knowledge.

In this paper, we introduce **Ne**uro-**Sy**mbolic model for **Mo**tion **F**orecasting (NeSyMoF), a novel architecture designed to leverage the strengths of both deep learning and symbolic reasoning in motion forecasting. NeSyMoF generates logical rules that condition the path-generation process, offering a transparent rationale for its predictions. NeSyMoF includes a novel neuro-symbolic reasoning module (NSR) that makes use of logic tensor networks to learn a grounding of predicates that describe the observed and future maneuvers performed by a dynamic agent in the scene in an end-to-end manner. Using an implication operation that links observed maneuvers with potential future maneuvers, NeSyMoF generates logical, interpretable paths. In addition to providing explanations for path predictions, the logical rules represent means to detect hard scenarios for the model, where the rules and path are inconsistent. We evaluate our model with the Argoverse benchmark for path forecasting, as it includes challenging driving situations for multiple agent types and situations [15], necessary to extensively evaluate our model. The results of our evaluation show that NeSyMoF outperforms state-of-the-art interpretable models for single-mode predictions while providing logic-based explanations

*Telecooperation Lab, Technical University of Darmstadt, Germany {doula, max, sanchez}@tk.tu-darmstadt.de.

for its forecasts, that articulate the reasoning behind predictions, making NeSyMoF more adapted for human-centric applications.

II. Related Work

A. Multi-Modal Motion Forecasting.

The advances in deep learning have paved the way for neural networks to become the standard for statistical trajectory forecasting. Several approaches rely on Long Short-Term Memory (LSTM) networks [16] combined with interaction features, such as social forces [17] and social pooling [18], to incorporate the mutual influence of dynamic agents present in the scene on the future paths. However, the unique consideration of social interactions without the spatial context may lead to invalid paths, such as cars driving in areas allocated for pedestrians. To mitigate these limitations, several works use multichannel rasterized images as inputs to convolutional layers [19] or transform it to a graph of lanes [6] to serve as spatial priors. In addition to the effective capturing of the agent's spatial context, motion forecasting approaches need to account for the multitude of possible future paths that a dynamic agent may take, given its current state. Multi-modal motion forecasting has been addressed in the literature using several techniques, such as graph neural networks [6], [20] and attention mechanisms [19], [21], [22]. Approaches as in [23], [4], [24] use variational auto-encoders (VAE) or generative adversarial networks (GAN) to learn a distribution of possible paths conditioned by scene elements and past agent states. The prediction process of these models consists of sampling multiple future paths from the learned latent distributions. The major drawback is the difficulty of interpreting the outputs of such models and linking them to human-understandable intentions. Several approaches address the interpretability problem using different techniques, such as maneuver classification [9], [10], goal and intention prediction [12], [5], [25], [26], and social interaction cues [27], [28]. In this paper, we keep the flexibility offered by previous multi-modal generation techniques while providing human-understandable interpretations in the form of definite logic-based rules for the generated paths.

B. Neuro-Symbolic Architectures

Neuro-symbolic architectures are designed to harness the strength of neural networks in extracting complex patterns and features from data and integrate this with the structured and clear representations offered by symbolic reasoning. This combination facilitates the execution of logic-based reasoning processes that are interpretable by humans, thereby bridging the gap between advanced computational models and intuitive human understanding. The overarching goal is to create systems that not only learn from vast amounts of data but also reason in ways that align with human logic, making the outcomes of such systems more transparent and trustworthy. Neuro-symbolic models have found applications in several tasks, such as semantic image interpretation [13], cognitive robotics [29], and generating programs for language-guided robot manipulation [30]. For autonomous driving, neuro-symbolic models have been used to generate optimal navigation programs that are human interpretable [14].

In this paper, we use neuro-symbolic models to create understandable rules for the path prediction process, making it ideal for applications focused on human users where clarity is essential.

III. NeSyMoF: Neuro-Symbolic Motion Forecasting

A. Problem Formulation

We consider the task of multi-modal motion forecasting from past trajectories and spatial context information. We consider a scene $\mathcal{S} = \{\mathcal{A}, \mathcal{M}\}$. $\mathcal{A} = \{A_i\}_{i\in[0,N]}$ denotes the set of N agents of interest present in the scene. Each agent A_i is represented by the history of its $2D$ positions $\mathcal{X}_i^{past} = (x_i^t, y_i^t)_{t\in[0:T_{obs}]}$ over an observation time T_{obs}. $\mathcal{M}$ denotes the set of M consecutive lane nodes extracted from a high-definition map (HD map) that represents the environment. Each node j has F features and is represented by the vector $f_j \in \mathbb{R}^F$. We refer to the feature matrix for all nodes in the scene as $X \in \mathbb{R}^{M\times F}$. We follow [6] and model the lanes as a graph with M nodes and represent the node connections with 4 adjacency matrices $\{K_j\}_{j\in\{pre,suc,left,right\}}$, where $K_j \in \mathbb{R}^{M\times M}$. Our goal is to learn a parametrized function that predicts the future motion $\mathcal{X}_i^{future} = (x_i^t, y_i^t)_{t\in[T_{obs}:T_{pred}]}$ of every agent A_i, where T_{pred} is the prediction horizon.

B. Model Architecture

In this section, we introduce the architecture of NeSyMoF, our neuro-symbolic motion forecasting model. As depicted in Fig 1, our network is composed of 4 main blocks:(1) a past path encoder, (2) a future path encoder, (3) a neuro-symbolic reasoning block (NSR), and (4) a multi-modal path decoder with a scoring head to assign probabilities to the generated paths. NeSyMoF follows a conditional variational auto-encoder architecture (CVAE) [31], where the posterior distribution of future paths $P(\mathcal{X}_i^{future}|\mathcal{X}_i^{past})$ is learned with the help of the latent variable Z.

C. Past Path Encoder

The past path encoder block encodes features that describe the observed agent environment between T_0 and T_{obs}. The enoding of past paths is performed using 3 subnetworks. First, an **agent encoder** extracts features from the observed path $\mathcal{X}_i^{past}$ of the agent of interest A_i. Second, a **lane encoder** extracts features from the graph representation of the HD map. Third, an **agent-environment interaction** to model interactions between A_i, the k nearest agents, and the lane graph.

1) Agent Encoder: the agent encoder is a 1-D convolution layer followed by a feature pyramid network g_θ, for which input is $\mathcal{X}_i^{past}$. The agent encoder outputs an embedding vector e_i^{past} according to Eq. 1

$$e_i^{past} = g_\theta(CNN(\mathcal{X}_i^{past}; W_{AE})) \tag{1}$$

where W_{AE} are shared weights among all agents in the scene.

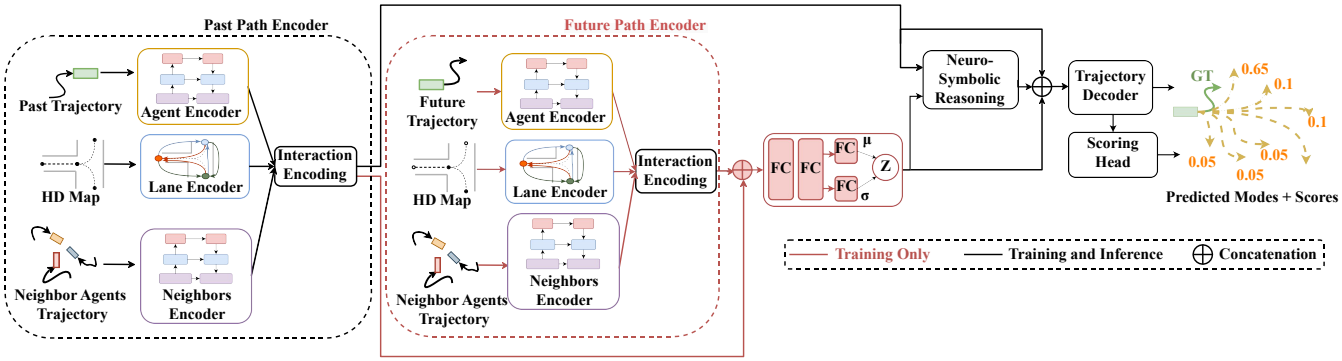

Fig. 1. NeSyMoF Model Architecture.

2) Neighbors Encoder: The neighbors encoder extracts features from the trajectories of the nearest agents around the agent of interest A_i. These features are encoded using a convolutional neural network, similar to the agent encoder, to produce embedding vectors for the neighboring agents. The neighbor embeddings e_j^{past} are computed by applying Eq. 2.

$$e_j^{past} = g_\theta(\text{CNN}(X_j^{past}; W_{NE})) \tag{2}$$

where W_{NE} represents shared weights among all neighboring agents in the scene.

a) Lane Encoder: To encode the lanes, we follow [6] and use two *LaneGCN* residual blocks. The lane features Y are obtained by applying Eq. 3.

$$\begin{aligned} Y =& XW_0 + \sum_{j\in\{left,right\}} K_j X W_j + \\ & \sum_c K_{pre}^c XKW_{pre,c} + K_{suc}^c XKW_{suc,c}, \end{aligned} \tag{3}$$

where $c \in \{1, 2, 4, 8, 16, 32\}$ denotes the dilation sizes of the *LaneGCN* residual blocks.

3) Interaction Encoding: We use multi-layer attention networks to capture the global interaction between the agent of interest, the lane segments, and the k nearest agents, as described in Eq. 4

$$\begin{aligned} E_i =& SelfAttn(e_i^{past}) + CrossAttn(e_i^{past}, Y) + \\ & CrossAttn(e_i^{past}, e_{j\neq i}^{past}), \end{aligned} \tag{4}$$

with $SelfAttn$ and $CrossAttn$ defined as:

$$CrossAttn(e_i^{past}, Q) = Softmax(\frac{Qe_i^{past.T}}{\sqrt{d_k}})e_i^{past} \tag{5}$$

$$SelfAtt(e_i^{past}) = Softmax(\frac{e_i^{past}e_i^{past.T}}{\sqrt{d_k}})e_i^{past}, \tag{6}$$

where $Q \in \{Y, e_{j\neq i}^{past}\}$ for our case and d_k is the dimension of the key vectors.

D. Future Path Encoder

The FPE block adopts the same architecture used to encode past paths. The learned features e_i^{future} are then concatenated with e_i^{past} and passed to a fully connected network. The goal of the fully connected network is to learn a latent Gaussian distribution $\mathcal{N}(\mu, \sigma)$, from which we sample the latent variable $P(z|e_i^{past}, e_i^{future})$.

E. Neuro-Symbolic Reasoning Module

The goal of the NSR module is to learn interpretable logic rules that describe the relationship between the observed path $\mathcal{X}_i^{past}$ of an agent A_i and its future path $\mathcal{X}_i^{future}$. For this, we define a first-order language (FOL) $\mathcal{L}$ with a signature $\Sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$, where $\mathcal{C}$ is a set of constants, $\mathcal{F}$ a set of functions, and $\mathcal{P}$ a set of predicates. For our application, $\mathcal{C} = \{\mathcal{X}_i\}$ represents the set of path identifiers, $\mathcal{F} = \emptyset$, and $\mathcal{P} = \{\mathcal{P}_{observation}, \mathcal{P}_{maneuver}\}$. $\mathcal{P}_{observation} = \{\texttt{Past}, \texttt{Future}\}$ is the set of two predicates indicating if the portion of the path is the observed one (i.e., $\mathcal{X}_i^{past}$) or the predicted one (i.e., $\mathcal{X}_i^{future}$). And, $\mathcal{P}_{maneuver} = \{\texttt{Accelerate}, \texttt{Decelerate}, \texttt{Uniform}, \texttt{Forward}, \texttt{Right}, \texttt{Left}, \texttt{LaneChange}\}$ is the set of predicates that indicates the maneuver performed during a portion of a path. we have deliberately chosen to represent agent maneuvers with a set of 7 predicates, directly reflecting the maneuvers present large path prediction benchmarks such as the Argoverse dataset [15]. This selection was driven by the goal of achieving a match between our model's output and the dataset's actual maneuver types. However, the set of predicates can be easily extended. To obtain the predicates, we perform the path clustering technique in [9], which determines the label of the path based on the deviation of the agent's trajectory throughout the time steps with respect to the scene. The FOL formulas based on Σ can describe simple facts about the learned trajectories and the logical relationships between the observed part (i.e., $\mathcal{X}^{past}$) and the parts to be predicted by our neural network (i.e., $\mathcal{X}^{future}$) through an implication relationship. The goal of the reasoning module is to learn rules of the form:

$$\begin{aligned} \forall \mathcal{X}_i(\texttt{Past}(\mathcal{X}_i) \wedge \texttt{PastManeuver}(\mathcal{X}_i) \rightarrow \\ \texttt{Future}(\mathcal{X}_i) \wedge \texttt{FutureManeuver}(\mathcal{X}_i)) \end{aligned} \tag{7}$$

Under the form formalized in Eq. 7, it is possible to link the observed state and maneuver of the agent to possible future maneuvers that can be implied from the observations. Since we consider multi-modal predictions, the NSR module

learns a rule for each predicted path, separately, based on the observed state and maneuver performed by the agent, which is jointly considered by all modes. An example of such rules is described in Eq. 8.

$$\forall \mathcal{X}_i(\texttt{Past}(\mathcal{X}_i) \wedge \texttt{Right}(\mathcal{X}_i) \rightarrow \texttt{Future}(\mathcal{X}_i) \wedge \texttt{Decelerate}(\mathcal{X}_i)) \tag{8}$$

The rule described in Eq. 8 is interpreted as follows: if the observed path indicates a turn to the right, then the agent will decelerate in the future. In addition to providing explanations for path predictions, such logical rules represent means to detect hard scenarios for the model, where the rules and path are inconsistent. Fig 2 illustrates an example of a path prediction scenario, where each predicted mode is linked to a rule generated by the NSR module. To learn the predicates, we consider logic tensor networks (LTN) [32], where MLPs are used to learn a grounding $\mathcal{G}$ for each element of Σ. To define the semantics of our FOL, we use the real logic operators in [32], as they are better suited for gradient-descent optimization. The NSR module consists of k LTN blocks to learn the rule for each of the k paths that the model generates. As depicted in Fig. 3, each LTN block consists of 2 subnetworks with $softmax$ outputs to learn a grounding for each of the 6 maneuver predicates for the past and future paths. The implication operation outputs a satisfiability level for the correctness of the linking between the past and future maneuvers with the highest probability. As such, the NSR module enhances interpretability by generating logical rules that explain predicted trajectories based on classified maneuvers, providing human-understandable insights into the model's decision-making process.

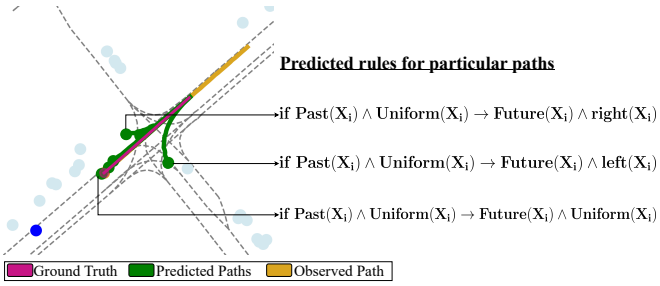

Fig. 2. An illustrative example of predicted paths and Rules.

F. Trajectory Decoding and Path Scoring

The trajectory decoder receives as input a concatenation of the PPE, FPE, and NSR outputs and forwards them to a regression module that predicts K future paths using a residual network and a linear layer. The generated trajectories and the output of NSR module are forwarded to a scoring head that outputs K confidence scores, $\{c_k\}_{k\in[0,K]}$, for each generated path. For the calculation of the score, the trajectories and the NSR predicates are encoded using two 2-layer MLPs. The generated features are then passed to an attention layer followed by a Softmax layer to output confidence scores for each trajectory.

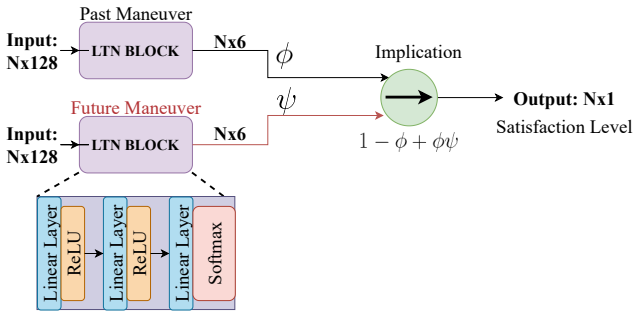

Fig. 3. Architecture of an LTN block in the neuro-symbolic reasoning module (NSR).

G. Learning

The model is trained end-to-end since all modules are differentiable. We use a combination of four losses as indicated in Eq. 9

$$L = L_{goal} + L_{wp} + L_{CVAE} + L_{LTN} \tag{9}$$

L_{goal} is a max-margin loss, for which the goal is to enable the network to learn to assign high probability scores to paths that same final position or goal as the ground truth path. To calculate L_{goal}, we determine, k', the path with the smallest final position displacement with respect to the ground truth, out of the K generated paths. The max-margin loss is then calculated as described in Equation 10.

$$L_{goal} = \frac{1}{(K-1)} \sum_{k \neq k'} \max(0, c_k + \varepsilon - c_{k'}) \tag{10}$$

The goal of the $L_{wp,k}$ loss is to enable the model to learn possible waypoints that an agent can take in each of the timesteps, for each mode.

$$L_{wp,k} = \frac{1}{|T_{pred} - T_{obs}|} \sum_{t=T_{obs}}^{T_{pred}} d(\mathcal{X}_{t,k}^{future} - \mathcal{X}_t^*) \tag{11}$$

where $\mathcal{X}_{t,k}^{future}$ is the predicted position of the $k-th$ mode at time step t, $\mathcal{X}_t^*$ is the ground truth position at time step t, and $d(x)$ is the smooth l_1 loss defined as:

$$d(x_i) = \begin{cases} 0.5x_i^2 & \text{if } \|x_i\| < 1 \\ \|x_i\| - 0.5 & \text{otherwise,} \end{cases} \tag{12}$$

L_{CVAE} represents the loss for the CVAE, consisting of a reconstruction term and a Kullback-Leibler divergence term, encouraging the encoder to produce latent variables that follow a distribution close to the standard normal distribution, as shown in Equation 13.

$$L_{CVAE} = \text{NLL}(e_i^{future}|z, e_i^{past}) + D_{KL}(P(z|e_i^{past}, e_i^{future})||\mathcal{N}(0, I)) \tag{13}$$

For the loss of our neuro-symbolic reasoning module, We use the satisfiability, a_i, of the implication operation as a training objective for the LTN module, meaning that we are expecting a_i to be 1 if the implication is correct. The LTN

loss is a root-mean-square loss combined with two cross entropy terms for the past and future maneuver classification, as expressed in Eq. 14

$$L_{LTN} = (\frac{1}{N}\sum_{i=1}^{N}(1-a_i)^2)^{\frac{1}{2}} - \sum_{i=1}^{N} c_i log(\phi_i) - \sum_{i=1}^{N} c_i log(\psi_i) \tag{14}$$

where c_i is the true maneuver class, and ϕ_i and ψ_i are the output of the past and future maneuver grounding blocks, respectively.

IV. Evaluation

To evaluate NeSyMoF, we, first, compare our approach to state-of-the-art models that claim interpretability, to assess predictive capabilities of NeSyMoF. Second, we conduct an ablation study to evaluate the influence of the NSR module on the prediction of interpretable paths.

A. Experimental Setting

B. Dataset

We use the Argoverse dataset for motion forecasting [15] to evaluate our model, as it includes challenging driving situations, necessary to extensively evaluate our model. The dataset has over 30K scenarios recorded as a sequence of frames with the positions of every present agent in 2D coordinates. The task consists of predicting the position of agents in a 3-second horizon, given a 2 seconds-long observation time-lapse. Furthermore, the dataset provides HD map information that we use to learn the lane representations.

C. Metrics

We use three commonly used metrics for motion forecasting: *minADE*, *minFDE*, and *MR*. The metric *minADE* denotes the minimal average displacement error, defined as the average l_2 distance between the predicted and the ground truth paths over $K = 6$ predictions, as described in Equation 15.

$$minADE = \min_k \frac{1}{|T_{pred} - T_{obs}|} \sum_{t=T_{obs}}^{T_{pred}} \|\mathcal{X}_{t,k}^{future} - \mathcal{X}_{t,k}^{*}\|_2 \tag{15}$$

where $|T_{pred} - T_{obs}|$ is the total number of time steps, $\mathcal{X}_{t,k}^{*}$ is the ground truth position at time t, and $\mathcal{X}_{t,k}^{future}$ is the k-th predicted position at time t, and the minimum is taken over all K predictions. The *minFDE* denotes the minimal final displacement error and is defined as the l_2 distance between the predicted and the ground truth paths at the last step, as defined in Equation 16.

$$minFDE = \min_k \|\mathcal{X}_{t,k}^{future} - \mathcal{X}_{t,k}^{*}\|_2 \tag{16}$$

Lastly, *MR* refers to the *Miss Rate* value, representing the ratio of predictions located more than 2.0 meters away from the ground truth, as defined in Equation 17.

$$MR = \frac{\text{Number of times minADE} > 2m}{\text{Total number of predictions}} \tag{17}$$

D. Implementation

For implementing Logic Tensor Networks in the NSR module, we use PyTorch version 1.9.0 and the LTNtorch library[1] as documented in [32]. The coordinate framework is initialized with the agent's 2D position at $t = 0$ at its center, directing the positive x axis towards the agent's location at $t = 1$. Training occurred on a *TESLA V100* GPU, adopting a batch size of 64 throughout 102 epochs, and utilizing the Adam optimizer. An initial learning rate of 10^{-3} is set, with a decay rate of 10^{-5} applied every 20 epochs. We derive the ground truth labels for the predicates following the approach in [9], applying constrained K-means clustering to categorize maneuvers. While the clustering in [9] considers the full path, we subdivide each path into an observed and predicted fragment and classify the fragments separately using the same process. Each path fragment is allocated into one of the predetermined maneuver classes $\mathcal{P}_{maneuver} = \{\texttt{Accelerate}, \texttt{Decelerate}, \texttt{Uniform}, \texttt{Forward}, \texttt{Right}, \texttt{Left}, \texttt{LaneChange}\}$.

E. Comparison to State-of-the-Art

We compare our approach to two baseline approaches provided by the Argoverse benchmark and to a number of state-of-the-art approaches from the Argoverse leaderboard that claim interpretability in their papers.

For $K = 1$, NeSyMoF outperforms all the models, exceeding the Argoverse baselines by approximately 49% in minADE, 51% in minFDE, and 30% in MR, indicating significant improvements across all metrics. Compared to other reported interpretable approaches, NeSyMoF outperforms by up to 18.89% in minADE, 21.41% in minFDE, and 12.86% in MR (numbers relative to TNT [5]). Out of the interpretable models that we use for our comparison, mmTransformer [22] represents the best-performing model. NeSyMoF outperforms mmTransformer in all metrics, for $K = 1$. We notice, that the difference between NeSyMoF and mmTransfomer is more apparent in the minFDE metric than minADE and MR. The higher performance gain in minFDE is expected due to the global prior provided by the NSR module, which influences the overall direction and final goal of the predicted path and is more reflective in the minFDE metric.

For $K = 6$, NeSyMoF achieved the second-best results across all metrics, slightly trailing mmTransformer. NeSyMoF's reliance on a single observed trajectory as ground truth may limit its ability to generalize across multiple potential future paths.

The interpretable state-of-the-art approaches leverage different cues to provide interpretations for the path prediction process, such as region-based training [22], goal-conditioned path prediction [5], and road segment-conditioned path prediction [33]. The neuro-symbolic logical rules generated by NeSyMoF provide a more interpretable framework than goal-conditioned path generation and region-based learning

[1]LTNtorch: https://github.com/tommasocarraro/LTNtorch

Model	K=1			K=6		
	minADE	minFDE	MR	minADE	minFDE	MR
Argoverse Baseline (LSTM) [15]	2.96	6.81	0.81	2.34	5.44	0.69
Argoverse Baseline (NN) [15]	3.45	7.88	0.87	1.71	3.29	0.54
WIMP [33]	1.82	4.03	0.62	0.90	1.42	0.16
TNT [5]	2.17	4.95	0.70	0.90	1.44	0.16
SAMMP [34]	1.81	4.08	-	0.95	1.55	0.19
AutoBot [35]	1.83	4.10	0.63	0.89	1.41	0.16
mmTransformer [22]	1.77	4.00	0.62	**0.84**	**1.33**	**0.15**
NeSyMoF (ours)	**1.76**	**3.89**	**0.61**	0.88	1.39	0.16

TABLE I

RESULTS ON ARGOVERSE MOTION FORECASTING BENCHMARK. THE BEST RESULTS ARE MARKED IN BOLD. THE SECOND-BEST RESULTS ARE UNDERLINED.

Model	K=1			K=6		
	minADE	minFDE	MR	minADE	minFDE	MR
NeSyMoF (no NSR)	1.68	3.73	0.65	0.80	1.15	0.13
NeSyMoF	**1.42**	**3.10**	**0.50**	**0.74**	**1.10**	**0.10**

TABLE II

ABLATION STUDIES ON ARGOVERSE MOTION FORECASTING BENCHMARK

because they explicitly articulate the reasoning behind predictions using understandable, human-like language, bridging the gap between complex data patterns and intuitive explanation, making NeSyMoF more suitable for real-life applications with non-expert users, such as users of intelligent vehicles.

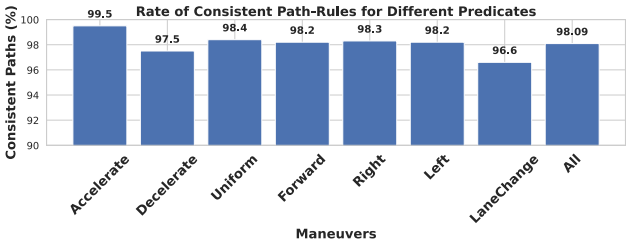

Fig. 4. Rate of consistent path-rules for the different predicates.

F. Ablation Study

We conduct an ablation study to investigate the effect of the neuro-symbolic reasoning module on the performance of the network. We follow [33], [5], [9] and perform our ablation study on the validation split of the Argoverse benchmark with and without the NSR module. The results in Table II show that the NSR module improves the prediction performance in all metrics for the single mode prediction $K = 1$, outperforming the model with no NSR by approximately 15.48% in minADE, 16.89% in minFDE, and 23.08% in MR, indicating significant improvements across all metrics. This result demonstrates that the learned logical rules improved the prediction of future paths by serving as logical priors. Fig 5 demonstrates qualitative examples of generated paths by NeSyMoF with their logical interpretations. By learning possible logical combinations between past and future maneuvers, the model generates valid future paths with valid logical interpretations. For the multi-modal prediction case ($K = 6$), the NSR module contributes to the overall performance of NeSyMoF, leading to a better performance by 7.5% in minADE, 4.35% in minFDE, and 23.08% in MR. However, the influence of the NSR module is less apparent compared to $K = 1$. We attribute this behavior to the lack of multi-modal ground truth in path forecasting benchmarks, which leads to a limited contribution of the NSR module for the multi-modal case since during our training we use the same predicates for all 6 NSR rules as they represent the unique ground truth we could use. Despite this limitation, the NSR module can generate multiple descriptions for a single observed path and link them to the corresponding generated paths, as depicted in the examples in Fig 5. Furthermore, the generated paths remain consistent with the generated rules, highlighting the capability of the NSR module to describe the predicted paths correctly and its capability to act as a strong prior for the generation process. In Fig 4, we report the percentage of consistent paths with respect to the logical rules for every generated path for $K = 6$, using the validation split of the Argoverse benchmark, after training the model with the full training split. We use the validation split, as we do not dispose of the ground truth data of the test split. Fig 4 shows that 98.09% of the generated paths are consistent with the generated rule, with rates ranging from 98.5% for the `Accelerate` predicate to 96.6% for `ChangeLane`.

G. Qualitative Results

In Fig 5, we provide a detailed qualitative analysis of the trajectories forecasted by NeSyMoF. For comparative purposes, we selected 4 distinct scenarios from the validation dataset due to the inaccessibility of ground truth for the test set. Each scenario illustrates the observed trajectory (in orange), and the actual future trajectory (in purple), alongside six forecasted trajectories (in green) on top of the lane maps provided by the dataset. Additionally, we demonstrate the logical rules corresponding to the top-3 forecasted trajectories based on their probability scores. The predictions in Fig 5 demonstrate the capability of NeSyMoF

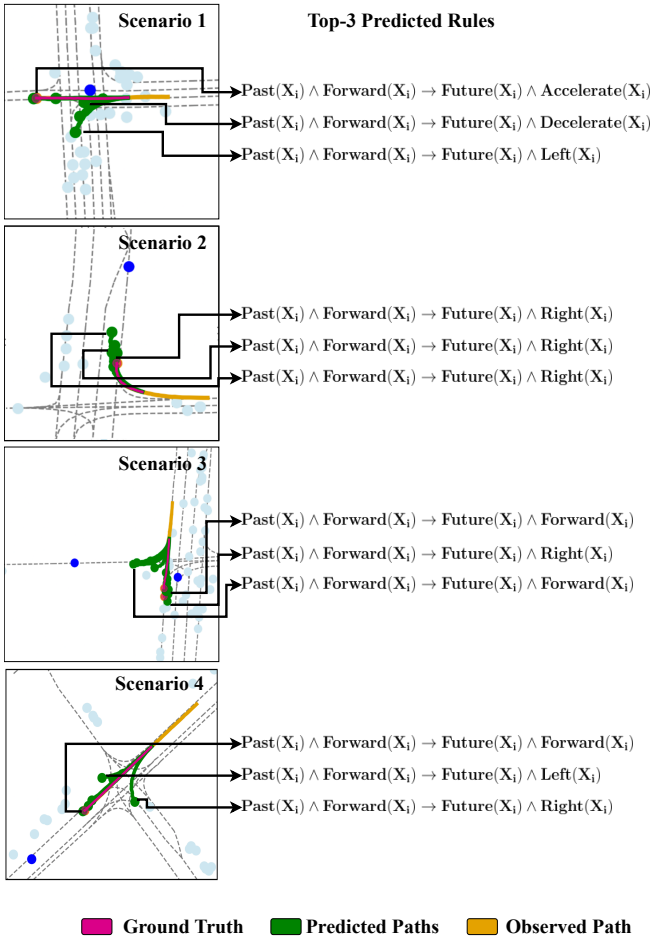

Fig. 5. Example of qualitative results of predicted paths and Rules on different scenarios from the Argoverse dataset.

in integrating spatial context from the map data, underscoring its adeptness in assimilating lane information to inform and condition the trajectory forecasting process. The superposition of the ground truth path and at least one of the predicted paths shows that NeSyMoF is capable of generating paths that are conformal to the ground truth. This demonstrates the capability of the CVAE module to capture the data distribution and the capability of the scoring head to correctly assign higher probabilities to paths that resemble the ground truth data. The generated logical rules correctly describe the generated rules, demonstrating the capability of the NSR module to generate descriptions that are consistent with the generated paths, offering human users an interpretable interface to analyze better the behavior of the model.

V. Discussion and Limitations

We are confident that NeSyMoF contributes important insights into the integration of neuro-symbolic approaches in the design of interpretable path prediction models. Compared to other interpretable models such as Multipath++, which primarily focus on semantic classification of paths, NeSyMoF provides a more explicit and human-like interpretability. The logical rules generated by our NSR module not only categorize maneuvers but also explain the reasoning behind each prediction, making the decision-making process transparent and more aligned with human logic. The interpretability provided by NeSyMoF offers important practical implications. By generating clear, logical explanations for predicted paths, the model enables operators to better understand and anticipate potential risks, make informed decisions, and adjust plans in real time to avoid collisions. This level of transparency is essential for improving safety in autonomous systems and building trust with human users.

In designing the first-order logic language for our model, We have chosen the predicates based on the common maneuvers present in large path prediction benchmarks such as the Argoverse dataset [15], even though the flexibility of our framework allows for the integration of additional predicates to capture a wider array of maneuvers. For instance, the inclusion of a predicate for `backward` driving, while not represented in the Argoverse dataset, could enhance the model's descriptive power for scenarios like parking or reversing, which are common in real-world driving but absent in the current data. Such an extension would enable the model to characterize an even broader spectrum of behaviors.

As we demonstrate in Sections IV-F and IV-G, the NSR module is capable of providing consistent rules with the generated paths. However, this consistency can be violated due to the under-representation of rare maneuvers and the statistical nature of our model. While the cases, where the generated paths do not correspond to the generated rules, are rare (1.91%, see Fig 4), this inconsistency can be leveraged to estimate the uncertainty of the model and highlight scenarios that are difficult for the model to learn. The identification of such scenarios is the core of different learning paradigms such as active learning and curriculum learning, where difficult scenarios are identified based on semantic inconsistencies in the model predictions and prioritized during the learning process. While such approaches do exist for perception tasks [36], approaches that consider path prediction from this perspective are rare. We believe that the nature of our NSR module offers a considerable starting point for research in that direction.

VI. Conclusion

In this paper, we present NeSyMoF, a novel neuro-symbolic model for multi-modal path forecasting. NeSyMoF generates first-oder logic rules that describe the predicted paths. Furthermore, we propose the first active learning strategy that semantic consistency between the first-order logic rules produced by NeSyMoF and the generated paths to prioritize difficult scenarios during the training. The results of our evaluation demonstrate the effectiveness of NeSyMoF in comparison with interpretable state-of-the-art models while being able to generate clear logical interpretations of the generated paths, making NeSyMoF more adapted for human-centric applications.

ACKNOWLEDGMENT

This work has been funded by National Research Center for Applied Cybersecurity ATHENE and LOEWE initiative (Hesse, Germany) within the emergenCITY center.

REFERENCES

[1] X. Ren, T. Yang, L. E. Li, A. Alahi, and Q. Chen, "Safety-aware motion prediction with unseen vehicles for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 731–15 740.

[2] S. Yi, H. Li, and X. Wang, "Pedestrian behavior modeling from stationary crowds with applications to intelligent surveillance," *IEEE transactions on image processing*, vol. 25, no. 9, pp. 4354–4368, 2016.

[3] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 935–942.

[4] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.

[5] 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, pp. 895–904.

[6] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.

[7] E. A. Pool, J. F. Kooij, and D. M. Gavrila, "Using road topology to improve cyclist path prediction," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 289–296.

[8] M. Körber, "Theoretical considerations and development of a questionnaire to measure trust in automation," in *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018) Volume VI: Transport Ergonomics and Human Factors (TEHF), Aerospace Human Factors and Ergonomics 20*. Springer, 2019, pp. 13–30.

[9] P. Bhattacharyya, C. Huang, and K. Czarnecki, "Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving," *arXiv preprint arXiv:2206.14116*, 2022.

[10] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 4363–4369.

[11] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 759–776.

[12] P. Dendorfer, A. Osep, and L. Leal-Taixé, "Goal-gan: Multimodal trajectory prediction based on goal position estimation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[13] I. Donadello, L. Serafini, and A. d'Avila Garcez, "Logic tensor networks for semantic image interpretation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1596–1602. [Online]. Available: https://doi.org/10.24963/ijcai.2017/221

[14] J. Sun, H. Sun, T. Han, and B. Zhou, "Neuro-symbolic program search for autonomous driving decision module design," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 21–30.

[15] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[18] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[19] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[20] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9491–9497.

[21] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, "End-to-end contextual perception and prediction with interaction transformer," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5784–5791.

[22] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.

[23] M. Lee, S. S. Sohn, S. Moon, S. Yoon, M. Kapadia, and V. Pavlovic, "Muse-vae: multi-scale vae for environment-aware long term trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2221–2230.

[24] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1349–1358.

[25] Y. Yao, E. Atkins, M. Johnson-Roberson, R. Vasudevan, and X. Du, "Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1463–1470, 2021.

[26] Y. Hu, W. Zhan, L. Sun, and M. Tomizuka, "Multi-modal probabilistic prediction of interactive behavior via an interpretable model," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 557–563.

[27] L. Shi, L. Wang, C. Long, S. Zhou, F. Zheng, N. Zheng, and G. Hua, "Social interpretable tree for pedestrian trajectory prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2235–2243.

[28] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8823–8833.

[29] R. Chitnis, T. Silver, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic relational transition models for bilevel planning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4166–4173.

[30] K. Namasivayam, H. Singh, V. Bindal, A. Tuli, V. Agrawal, R. Jain, P. Singla, and R. Paul, "Learning neuro-symbolic programs for language guided robot manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7973–7980.

[31] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[32] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.

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

[34] 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.

[35] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal, "Latent variable sequential set transformers for joint multi-agent motion prediction," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[36] Z. Hu, X. Bai, R. Zhang, X. Wang, G. Sun, H. Fu, and C.-L. Tai, "Lidal: Inter-frame uncertainty based active learning for 3d lidar semantic segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 248–265.