Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Uncertainty-Aware Trajectory Prediction via Rule-Regularized Heteroscedastic Deep Classification

Kumar Manas[1], Christian Schlauch[2,3], Adrian Paschke[1,4],Christian Wirth[2] and Nadja Klein[3]

***Abstract*—Deep learning-based trajectory prediction models have demonstrated promising capabilities in capturing complex interactions. However, their out-of-distribution generalization remains a significant challenge, particularly due to unbalanced data and a lack of enough data and diversity to ensure robustness and calibration. To address this, we propose SHIFT (Spectral Heteroscedastic Informed Forecasting for Trajectories), a novel framework that uniquely combines well-calibrated uncertainty modeling with informative priors derived through automated rule extraction. SHIFT reformulates trajectory prediction as a classification task and employs heteroscedastic spectral-normalized Gaussian processes to effectively disentangle epistemic and aleatoric uncertainties. We learn informative priors from training labels, which are automatically generated from natural language driving rules, such as stop rules and drivability constraints, using a retrieval-augmented generation framework powered by a large language model. Extensive evaluations over the nuScenes dataset, including challenging low-data and cross-location scenarios, demonstrate that SHIFT outperforms state-of-the-art methods, achieving substantial gains in uncertainty calibration and displacement metrics. In particular, our model excels in complex scenarios, such as intersections, where uncertainty is inherently higher. Project page: https://kumarmanas.github.io/SHIFT/.**

I. Introduction

Trajectory prediction represents a fundamental challenge in autonomous driving and robotics, serving as a critical bridge between perception and planning systems [37, 22]. As illustrated in Fig. 1, autonomous vehicles must simultaneously reason about multiple possible future trajectories while accounting for road conditions, traffic rules, and multi-agent interactions. This complex decision-making process becomes particularly challenging in urban environments, where the interactions between vehicles, pedestrians, and infrastructure are fast-changing [25, 5]. Deep learning approaches for trajectory prediction have demonstrated remarkable potential in tackling these challenges [22]. Nevertheless, the safety-critical nature of autonomous driving introduces requirements that current state-of-the-art deep learning-based trajectory predictors struggle to address comprehensively.

First, models must generalize effectively from limited data, given the high cost of data collection and long-tail distribution of unique scenarios in urban environments. For example, most observed scenarios involve relatively simple behaviors, such as following a straight road, while rare safety-critical events, like near-accidents, are particularly challenging to capture due to their low frequency. Data efficiency of deep learning-based trajectory predictors has only recently gained attention in scientific investigations [16, 56].

Second, models must be robust to shifts in road conditions, perception systems, and geographic locations. Social norms and traffic rules provide valuable context for predicting compliant agent behavior [7]. However, informed or knowledge-guided deep learning-based trajectory predictors have predominantly focused on physical constraints [12, 1, 27, 55], leaving broader contextual prior knowledge underexplored [53].

Third, trajectory prediction involves multiple sources of uncertainty that must be carefully quantified. Epistemic uncertainty arises from model limitations and incomplete training data, while aleatoric uncertainty stems from the stochasticity of agent behavior and environmental dynamics [23]. Although most trajectory prediction models address the predictive multimodality resulting from these uncertainties (e.g., by framing it as a classification problem), many fail to provide sufficiently calibrated uncertainty estimates, leading to mode-collapse issues [2].

To address these requirements, we propose SHIFT (Spectral Heteroscedastic Informed Forecasting for Trajectories) to synergize uncertainty quantification with rules as soft constraints in a scalable framework as visualized in Fig. 2:

1) We extend the Heteroscedastic Spectral-normalized Gaussian Processes (HetSNGP) by Fortuin et al. [17] to the trajectory prediction using the CoverNet baseline architecture [39]. CoverNet frames the prediction as a classification problem. HetSNGPs enable simultaneous estimation of epistemic and aleatoric uncertainties through a distance-aware neural GP layer combined with input-dependent noise modeling, improving calibration and requiring minimal architectural change and computational overhead.
2) We model epistemic uncertainty while accounting for the inherent stochasticity of agent interactions through aleatoric uncertainty, enabling a sequential learning process for informative prior weight distributions. In stage 1, we encode traffic rules as synthetic training labels, capturing structured prior knowledge. These learned traffic rule priors are then imposed as regularization during stage 2, guiding model training on real-world observations. This approach soft-constrains trajectory

[1]Department of Mathematics and Computer Science, Freie Universität Berlin. [2]Continental Automotive Technologies GmbH, AI Lab Berlin. [3]Karlsruhe Institute of Technology, Scientific Computing Center, Methods for Big Data. [4]Fraunhofer Institute for Open Communication Systems, Berlin, Germany. Contact: kumar.manas@fu-berlin.de.

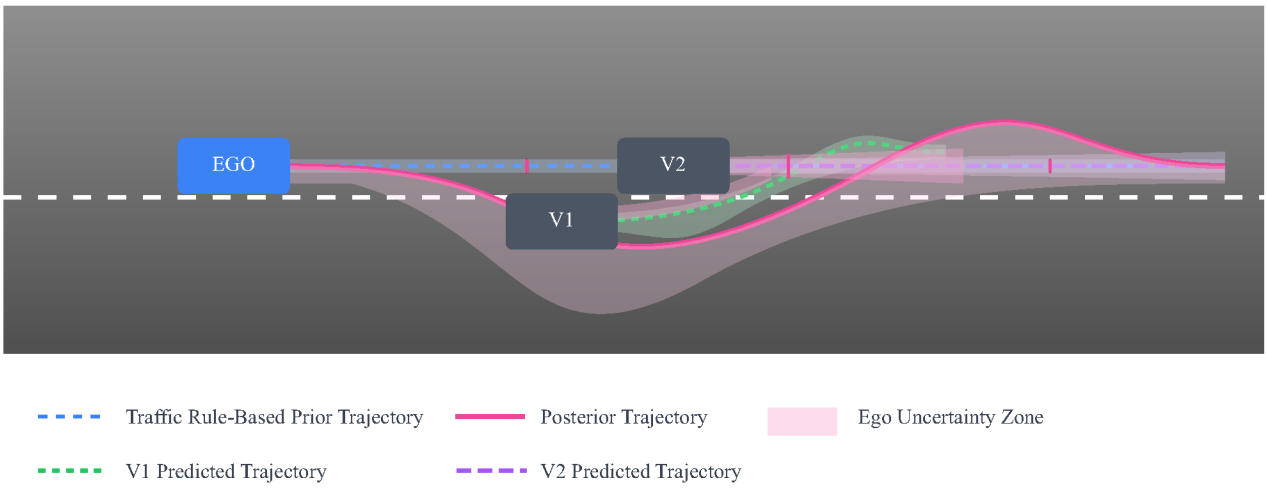

Fig. 1: Visualization of SHIFT's trajectory prediction framework. Rule-based priors (blue dashed) generate trajectories without interaction modeling, while interaction-aware HetSNGP posteriors (pink solid) produce uncertainty-calibrated ego paths. Surrounding vehicles' predictions (V1: green dashed, V2: purple dashed) with uncertainty regions (light green/purple shading) reveal multi-agent dynamics. Denser pink stripes (not size-based) highlight where vehicle interactions amplify prediction variance, showing how our model leverages static traffic rules with real-time uncertainty propagation across all actors.

predictions to align with traffic rules while preserving the flexibility to adapt to high-uncertainty and uncommon driving scenarios.

3) We employ a large language model (LLM) pipeline to semi-automatically generate synthetic training labels from natural language descriptions. This approach allows us to more easily scale the integration of multiple traffic rules, either as (a) a single "unified" prior learned from a single knowledge task or (b) a "chained" prior sequentially learned from multiple knowledge tasks in our first training stage.

SHIFT increases data efficiency and robustness by integrating sets of rules into the training process. Through extensive experiments on the nuScenes dataset [6], we demonstrate robust performance gains across varying training-set sizes (100% vs. reduced data) and challenging geographical splits (cross-location tests). Our results show particular strength in out-of-distribution scenarios, where the combination of uncertainty awareness and rule-based soft constraints provides more reliable predictions than conventional training methods.

II. Related Work

Trajectory Prediction Models: Existing models can be differentiated based on whether they predict the future behavior of all agents simultaneously (joint trajectory prediction) or a single agent at a time (marginal trajectory prediction), with most work focusing on the simpler latter problem [22, 54]. Deep learning approaches have achieved state-of-the-art performance in in-distribution evaluation settings, surpassing traditional methods such as kinematic models [48], Kalman filters and Bayesian Networks + [46]. Building on the work of Hagedorn et al. [22], deep learning-based trajectory predictors can be categorized based on their input representation, model backbone, and trajectory decoding methods. Input representations include birds-eye-view (BEV) image rasterizations [39], sparse vector encodings [50], graphs [57], and parametric curves [55]. Model backbone encompass convolutional neural networks (CNN) [39], recurrent neural networks [15], graph neural networks [28] and more recently transformers [50, 57, 36]. Trajectory decoding method include regression [36], anchor trajectory classification [39], trajectory refinement [8] or endpoint completion [20]. Our chosen baseline, CoverNet [39], utilizes a CNN backbone with BEV input rasterization and frames the prediction problem as anchor trajectory classification. This makes it particularly suitable for our modifications.

Informed Trajectory Prediction: Informed, or knowledge or rule-guided learning integrates explicit prior knowledge into the training process or architecture of deep learning models [53]. Most existing methods integrate physical knowledge to eliminate physically implausible outcomes, thereby simplifying the solution space [39, 12, 1, 27, 55]. In contrast, social norms and traffic rules act as soft constraints, as they can be violated in uncertain or atypical scenarios. Their integration often relies on transfer learning [4] or multi-task learning methods [41]. A related probabilistic approach learns informative priors sequentially from synthetic training labels, using them to regularize later training stages [44]. This setup offers flexibility, allowing sequences to be extended while minimizing catastrophic forgetting [14, 51]. However, unlike our method, it does not incorporate traffic rules and local rule constraints. Additionally, a significant challenge remains in translating natural language descriptions of rules into synthetic training labels. Recent efforts used retrieval augmented generation (RAG) [26] and large language models

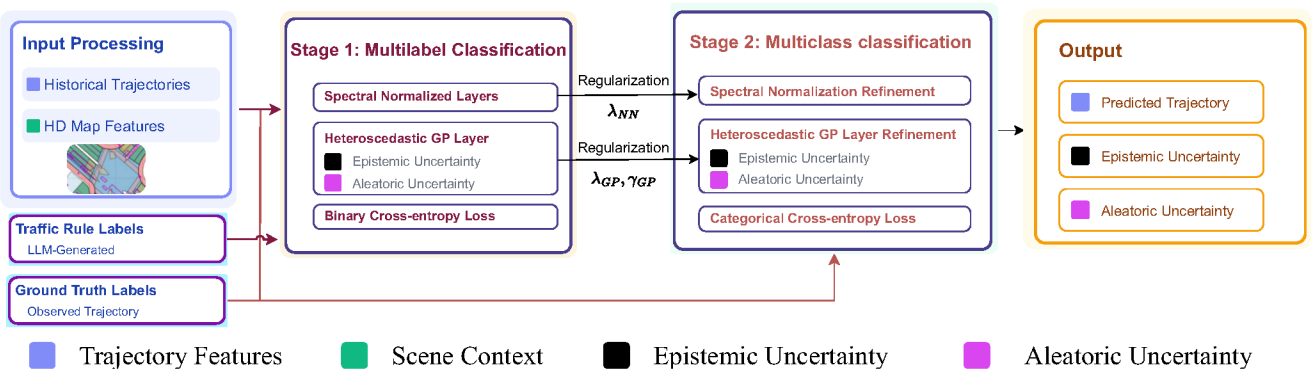

Fig. 2: Concept of the SHIFT framework for uncertainty-aware trajectory prediction. Our model is based on the CoverNet architecture, which frames the prediction as trajectory anchor classification. The backbone is spectral-normalized, and the output layer is replaced by a heteroscedastic Gaussian process. Given the processed input, comprising trajectory histories and map features, the model is trained in a multilabel classification on traffic rule labels generated from an LLM-based rule-filtering approach (stage 1). The obtained informative prior is used to regularize the multi-class classification training on ground truth labels from the observed data (stage 2). The model outputs consist of the predicted trajectories and disentangled epistemic and aleatoric uncertainty estimates. Best viewed in color.

(LLMs) to convert language descriptions into formal logic rules [21, 33, 34]. We employ these logic rules to filter input-dependent synthetic training labels from predefined anchor trajectories.

Uncertainty Quantification: Mode-collapse arises when trajectory prediction models fail to adequately capture the full diversity of possible futures [2]. This issue is commonly addressed through design choices in trajectory decoding [22]. For instance, in anchor trajectory classification, the softmax-normalized logits represent distributional information and identify the most likely top-k anchors [39]. However, deterministic deep learning models often exhibit overconfidence, which hinders their ability to accurately reflect the true variance and multiple modes of the underlying distribution. Their calibration can be substantially improved using principled uncertainty estimation techniques, such as those provided by approximate Bayesian deep learning [49]. Commonly used sampling-based methods, including Deep Ensembles [49], Monte Carlo (MC) Dropout [18], Stochastic Weight Averaging Gaussian (SWAG) [32], Laplace [13] or variational approximations [30], can be computationally prohibitive as they require multiple forward passes during prediction. Itkina and Kochenderfer [24] demonstrated the significance of estimating epistemic uncertainty in trajectory prediction through evidential deep learning, utilizing a classification-based prediction framework similar to ours. In contrast, Spectral-normalized Gaussian Processes (SNGPs) [29], belonging to the family of deterministic uncertainty estimators [9, 40], provide especially compute-efficient last-layer approximation. The Heteroscedastic Spectral-normalized Gaussian Process (HetSNGP) [17] leverages the heteroscedastic method for classification problems [10] to further improve calibration by disentangling the uncertainty into distance-aware epistemic and input-dependent aleatoric components [23]. Thus, unlike homoscedastic models, HetSNGP can tune the aleatoric uncertainty based on factors such as the complexity of the driving scenario at hand, instead of assuming a constant aleatoric uncertainty across all input conditions. This, in turn, can benefit the calibration of the epistemic uncertainty [11], which we employ for our rule-informed learning approach.

III. Methodology

We focus our discussion on the marginal trajectory prediction. Here, we aim to predict the future possible trajectories of a single agent at a time over some horizon, given the road topology, current state, and past trajectories of itself and its surrounding agents over some history, as well as the state of any traffic guidance systems. Our goal is to inform the model using traffic rules as explicit prior knowledge about the expected behavior of the agent. We propose SHIFT to approach rule informed marginal trajectory prediction in a scalable way, as shown in Fig. 2. To this end, SHIFT synergizes (A.) CoverNet as our baseline model for trajectory classification, (B.) HetSNGP as uncertainty-aware extension to CoverNet, that disentangles epistemic and aleatoric uncertainties, (C.) a regularization method which enables the integration of priors into our HetSNGP-CoverNet model, (D.) a sequential task setup to learn these priors from synthetic training labels, and (E.) an automated rule-filtering approach to encode these synthetic training labels from natural language descriptions. We describe these building blocks and (F.) our rule selection next.

A. Trajectory Classification with CoverNet

We adopt an anchor trajectory classification approach as the foundation for our framework. This approach simplifies the trajectory prediction by generating a input-dependent or input-independent set of anchor trajectories that capture all

plausible future motions [4, 8]. These anchors can be represented as classes in the output space $\mathcal{Y} = \{1, 2, \ldots, K\}$. The training data $\{(x_i, y_i)\}_{i=1}^N$ consists of samples x_i from the d-dimensional input space $\mathcal{X} \in \mathbb{R}^d$ and the training labels y_i being the classes in $\mathcal{Y}$ that most closely reflect the observed ground truth trajectories as measured by the Euclidean distance. The model can then be trained to classify the most likely anchor using a categorical cross-entropy loss.

Anchor trajectory classification offers several advantages. By appropriately sizing the anchor set, the prediction problem can be simplified, and the softmax-normalized logits provide probabilistic interpretations. Appropriately spaced anchors can also guard against drastic forms of mode-collapse [39]. However, the approach inherently introduces an irreducible approximation error, as the anchors may be arbitrarily distant from the actual ground truth trajectory. Consequently, the selection of anchor trajectories plays a critical role in the overall model performance.

For our baseline, we build on CoverNet [39] as it employs such an anchor classification framework and has been used in related work for informed trajectory prediction [4, 44, 45]. CoverNet processes bird's-eye-view (BEV), agent-centric rasterized input images using a ResNet-50 backbone. This rasterization preserves spatial relationships well. Next, CoverNet generates a predefined, input-independent anchor trajectory set based on all ground truth trajectories in the training data. Depending on a single distance parameter these trajectories are clustered to an appropriate number of anchors. This set of anchors is held fixed across all inputs. Note, that CoverNet also proposes an input-dependent anchor generation based on a dynamical bicycle model which encodes physical knowledge as hard constraint. Our proposed methodology is in principle agnostic to the kind of anchor sets and can be combined with such anchor generation for physical knowledge integration. For simplicity we focus on the predefined anchor set.

B. Uncertainty-Aware Trajectory Classification

We modify CoverNet as heteroscedastic Spectral-Normalized Gaussian Process (HetSNGP) [17]. The Spectral-Normalized Gaussian Process (SNGP), originally introduced by Liu et al. [29], combines the representative power of deep neural networks with uncertainty-aware Gaussian Process (GP) models [43]. The architecture consists of a deterministic, spectral-normalized feature extractor $f_{\theta_{\text{NN}}} : \mathcal{X} \to \mathcal{H}$, and an hierarchical GP output layer $f^{\text{L}}_{\theta_{\text{GP}}} : \mathcal{H} \to \mathcal{Y}$ (see Figure 3).

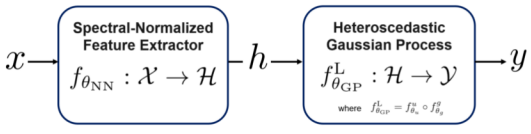

Fig. 3: Components of a HetSNGP model include a spectral-normalized feature extractor and a heteroscedastic Gaussian process as output layer.

The feature extractor is a neural network parameterized through θ_{NN} that maps the high dimensional input space $\mathcal{X} \in \mathbb{R}^d$ into a low dimensional hidden space $\mathcal{H} \in \mathbb{R}^m$ with $m \ll d$. The spectral-normalization enforces Lipschitz continuity, which can improve the generalization capability of the feature extractor [29] and prevent feature-collapse [52]. It is especially efficient to compute for the residual layers of a CNN-based backbone [19], as used in our CoverNet baseline. The resulting training data $\{(h_i, y_i)\}_{i=1}^N$ with its embedded inputs $h_i = f_{\theta_{\text{NN}}}(x_i)$ is then fed into the GP output layer.

The GP output layer employs a radial basis function (RBF) kernel κ and maps from the hidden space $\mathcal{H}$ into the output space $\mathcal{Y}$. Its hierarchical structure enables to quantify both, *aleatoric* and *epistemic* uncertainties. The GP latent variable g is modeled with a zero-mean multivariate normal distribution prior per-class $c \in \mathcal{Y}$:

$$g_c \sim \mathcal{N}(0, \kappa(h, h)) \quad \text{with} \quad g_c \in \mathbb{R}^{N \times 1}.$$

The posterior covariance matrix $\kappa(h, h)$ of g_c captures epistemic uncertainty. This uncertainty is class-independent, as the GP prior is shared across all classes.

To model aleatoric uncertainty, the HetSNGP introduces a hierarchical latent variable u with the following prior per-sample $i \in \{1, \ldots, N\}$:

$$u_i \sim \mathcal{N}(g_i, \Sigma(h, h)) \quad \text{with} \quad u_i \in \mathbb{R}^{1 \times K},$$

where $\Sigma(h, h)$ has a low-rank approximation based on a factor covariance matrix modeled as linear neural network with parameters θ_u [17] to achieve scalability with the number of classes K. The variable u captures input-dependent noise (e.g., sensor noise, agent intent ambiguity) and is processed through a heteroscedastic layer to produce aleatoric uncertainty. Unlike epistemic uncertainty, aleatoric uncertainty is sample-specific and varies across inputs. The hierarchical setup using both uncertainties correlates samples and classes in the posterior, leading to a better calibration [11].

The above formulation is computationally intractable. To achieve tractability the kernel is shared between all classes and approximated as $\kappa(h, h) = \Phi\Phi^\top$ using m random Fourier features (RFF) $\Phi_i = \sqrt{\frac{2}{m}} \cos(Wh_i + b)$, where W and b are fixed randomly sampled weights and biases respectively, based on Bochner's theorem [42, 29]. This RFF approximation reduces the computational complexity for inferring the posterior from $\mathcal{O}(N^3)$ to $\mathcal{O}(Nm^2)$ and allows us to write the GP as neural network layer with logits $g_c(h_i) = \theta_{g_c}\Phi_i$ and prior $\theta_{g_c} \sim \mathcal{N}(0, \mathbb{I})$. Since the posterior of θ_{g_c} is not of closed form, we use a Laplace approximation, which yields a Gaussian approximate posterior for the output weights θ_{g_c},

$$p(\theta_{g_c} | \{(x_i, y_i)\}_{i=1}^N) \approx \mathcal{N}(\theta_{g_c}; \theta^*_{g_c}, \Sigma_{g_c}),$$

where $\theta^*_{g_c}$ is the maximum a posterior (MAP) estimate and precision $\Sigma^{-1}_{g_c}$ is given by

$$\Sigma^{-1}_{g_c} \approx \mathbb{I} + \sum_{i=1}^{N} p_{i,c}(1 - p_{i,c})\Phi_i\Phi_i^\top, \tag{1}$$

where $p_{i,c}$ is the softmax output $p(y_i = c | u^*_{i,c})$ given the per-class per-sample $u^*_{i,c} = \theta^*_{g_c}\Phi^\top$. Overall, the trainable weights

in our output layer consist of $\theta_{\text{GP}} = \{\theta_u, \theta_{g_c} | \forall c \in \mathcal{Y}\}$. We adopt this output layer for our CoverNet baseline by replacing its dense output layer.

During inference, we approximate the marginal predictive distribution $p(y|h) = \int p(y|u)p(u|h)du$ of u using Monte Carlo (MC) sampling:

$$p(y \mid h) = \mathbb{E}_{p(\theta_{\text{GP}}|\{(h_i,y_i)\}_{i=1}^N)}\big[p(y \mid h, \theta_{\text{GP}})\big] \approx \frac{1}{S}\sum_{s=1}^{S} p(y \mid h, \theta_{\text{GP}}^{(s)}),$$

where S is the number of MC samples, and $\theta_{\text{GP}}^{(s)}$ are samples drawn from the approximate posterior distribution. Following Fortuin et al. (2021), a temperature parameter τ is introduced to recalibrate uncertainty at test time. The MC sampling is computationally efficient, as it involves sampling only from the output layer and supports parallelization.

C. Regularization using Informative Priors

To enable the integration of traffic rule as informative priors, we leverage the regularization method by Schlauch et al. [45]. The idea can be seen as an extension of online elastic weight consolidation [47] for GPs with RFF-approximated RBF-kernels. Assume we have some informative prior for the GP output layer $\pi = \mathcal{N}(\theta_{g_c}; \tilde{\theta}_{g_c}, \tilde{\Sigma}_{g_c})$ rather than a standard Gaussian prior as before. We then regularize the MAP estimate $\theta_{g_c}^*$ through this prior. This leads to the penalized log-likelihood

$$-\log p_{\theta_{g_c}}(y_i|h_i) - \frac{\lambda_{\text{GP}}}{2}(\theta_{g_c} - \tilde{\theta}_{g_c})^\top \tilde{\Sigma}_{g_c}^{-1}(\theta_{g_c} - \tilde{\theta}_{g_c}). \quad (2)$$

The posterior variance of θ_{g_c} is then given by

$$\Sigma_{g_c}^{-1} \approx \gamma_{\text{GP}}\tilde{\Sigma}_{g_c}^{-1} + \sum_{i=1}^{N} p_{i,c}(1 - p_{i,c})\Phi_i\Phi_i^\top. \quad (3)$$

This regularization scheme introduces two new hyperparameters λ_{GP} and γ_{GP}, which we assume are the same for all classes. The hyperparameter $\lambda_{\text{GP}} > 0$ tempers the overall prior, while the hyperparameter $0 < \gamma_{\text{GP}} < 1$ controls the learning decay when the model is sequentially trained [47].

In addition, assume we have some informative prior for the feature extractor $\pi = \mathcal{N}(\theta_{\text{NN}}; \tilde{\theta}_{\text{NN}}, \mathbb{I})$ too. Regularizing the feature extractor is then equivalent to $\mathcal{L}2$-regularization for the MAP estimates θ_{NN}^*, obtained by minimizing

$$-\log p_{\theta_{\text{NN}}}(y_i|h_i) - \frac{\lambda_{\text{NN}}}{2}(\theta_{\text{NN}} - \tilde{\theta}_{\text{NN}})^2, \quad (4)$$

which introduces an additional hyperparameter λ_{NN} controlling the strength of the parameter binding. Note, that we do not specifically place an informative prior on the parameters θ_u of the GP output layer as this head captures a property that is purely related to the data itself.

D. Learning Informative Traffic Rule Priors

We encode prior knowledge about traffic rules by incorporating additional tasks with synthetic training labels [44]. Given the input samples $x_i \in \mathcal{X}$ and the set of anchor trajectories, we define the synthetic training labels y_i as those classes in $\mathcal{Y}$ that yield rule-compliant anchor trajectories. In this multi-label classification setup, the model is trained to predict which anchors satisfy the respective traffic rule using a binary cross-entropy loss.

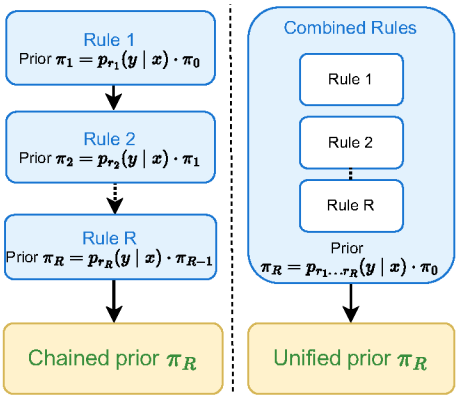

Fig. 4: Illustration of the possible task setups in the first training stage. The uninformed prior π_0 is either sequentially updated using distinct traffic rule labels p_{r_l} for each rule $l = 1, \ldots, R$ or updated once from a unified traffic rule $p_{r_1,\ldots,r_R}$ simultaneously. We denote the obtained final prior π_R as chained and unified priors, respectively. Both setups can be mixed. The obtained final prior is used to regularize the training on the observed ground truth trajectories in the next training stage.

Using an uncertainty-aware model in SHIFT, we can sequentially train on multiple tasks. Starting with an uninformative prior π_0, the posterior distribution from a previous task serves as informative prior π_l to regularize the subsequent task. The probabilistic regularization as described in Sec. III-C constitutes a key advantage over conventional transfer learning, which does not guard against catastrophic forgetting [14] or overly biasing subsequent training tasks [51]. Another important advantage lies in the flexible task setup, as visualized in Fig. 4. Thereby, we can define a single task that encodes prior knowledge about multiple traffic rules $p_{r_1,\ldots,r_R}(y|x)$ at the same time, learning a "unified prior" (right). Alternatively, we can define multiple tasks that encode prior knowledge about one traffic rule $p_{r_l}(y|x)$ each, learning a "chained prior" (left). This sequential task setup allows us to re-use priors easily, constituting an advantage over multi-task joint learning setups [41].

E. Synthetic Training Labels from Traffic Rules

To scale the integration of traffic rules, we employ a semi-automated filtering approach that (a) translates natural

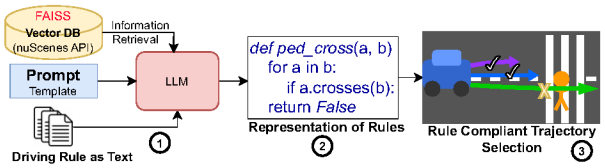

Fig. 5: High-Level LLM pipeline for synthetic training label creation:(1) rules, prompt, and nuScenes API are LLM inputs, (2) the LLM generates Python functions of rules and human experts review generated functions by running test cases using a sample traffic scene, (3) based on rules synthetic labels for training are generated.

language descriptions into executable Python functions using a LLM-based human-in-the-loop setup and (b) uses these functions to label each trajectory in an anchor set for rule compliance within a given scene. The first step (a) utilizes the retrieval-augmented generation (RAG) technique for the few-shot prompting of the LLM. We extended methodology developed by Manas et al. [33] using Llama3.3[1] model for this use case. When provided with a natural language rule description, RAG retrieves semantic elements from a dataset API. The LLM then integrates the relevant API calls to generate an executable Python function grounded in dataset API, which the user evaluates. If the function meets expert expectations, it is used in the second step (b) to label the rule compliance of the trajectory anchor set for all samples in the training dataset. Fig. 5 illustrates this process for synthetic label creation.

This pipeline accelerates rule integration, reducing the need for extensive manual coding or rule formalization. Various natural language traffic rules can be translated depending on the dataset API's capabilities and the available semantic information (e.g., lane types). However, even with sufficient semantic information, evaluating crucial temporal aspects (e.g., safety distances based on the velocities of multiple agents over time) remains challenging for the LLM. Further details on the LLM prompts and limitations are provided in Appendix C.

F. Rule Selection

We select two sets of rules to allow comparisons to previous work and demonstrate the scalability of SHIFT. These sets of rules model both globally-applicable and situational behavioral constraints, while avoiding the limitations of our LLM-powered rule-filtering approach. By integrating rules of varying complexity, we can account for more nuanced interactions with the traffic infrastructure.

Our first set contains a globally-applicable traffic rule:

- *Stay within the road boundaries*: This rule enforces the fundamental constraint of staying within drivable areas, reflecting high-level driving behavior.

Our second set of rules are situational stop-related traffic rules, including:

- *Stopping at red signals*: A rule that prohibits crossing traffic light zones when the red light is active, ensuring compliance with traffic signals.
- *Respecting right-of-way*: A rule that mandates yielding to other participants at yield or stop signs, capturing critical right-of-way interactions.
- *Prioritizing pedestrian safety*: A rule that requires giving way to pedestrians at active crossings, emphasizing safety in shared spaces.

IV. Experimental Design

We evaluate the model's performance in various scenarios, including standard full dataset training, low data regimes, and out-of-distribution generalization across different geographical locations within the nuScenes [6] dataset.

A. Dataset

We conduct our experiments on the nuScenes dataset [6]. We selected it for geographic diversity (Boston, USA, and Singapore), semantic information (including drivable areas, stop lines, walkways), and richly annotated agent trajectories. Its geographic split enables controlled out-of-distribution (OoD) evaluation, where models trained on data from one region (e.g., Boston's grid-like roads with right-side driving) are tested on another (e.g., Singapore's curved intersections and left-side driving). This mimics real-world deployment challenges where models encounter unseen environmental semantics. For our full dataset experiments, we utilize the train/val/test splits following Phan-Minh et al. [39]. In terms of location diversity, roughly 39% of the scenes are set in Singapore, while 61% take place in Boston. Refer to Appendix F for more details about the dataset.

B. Evaluation Metrics

We evaluate our approach using three categories of metrics: (1) Displacement Metrics, measuring spatial accuracy of predictions; (2) Distribution-aware Metrics, quantifying the quality of predicted probability distributions; and (3) Ranking Metrics, assessing the model's ability to assign higher probabilities to more likely trajectories. **minADE**$_k$ (Minimum Average Displacement Error): Measures the average l_2 distance in meters between the ground truth trajectory and the closest predicted trajectory among the top-k predictions. We report both minADE$_1$ and minADE$_5$ to evaluate single-mode and multi-mode prediction accuracy respectively. **minFDE** (Minimum Final Displacement Error): measures the minimum l_2 distance between the predicted final position and the ground truth final position among the top-k predictions. **Negative Log-Likelihood (NLL)**: Quantifies the quality of the predicted probability distribution by measuring the negative log-likelihood of the ground truth trajectory under the model's predictions. A lower NLL indicates better alignment between the predicted distribution and observed trajectories. **Expected Calibration Error (ECE)**: Measures the alignment between predicted probabilities and empirical frequencies [38]. ECE

[1]https://ollama.com/library/llama3.3

TABLE I: Comparison of SHIFT and baselines on Full Dataset of nuScenes. Our approach is compared against other uncertainty-aware classification baselines. Bold indicates best, and second best is underlined. Lower is better for metrics.

Model	$\text{minADE}_1 \downarrow$	$\text{minADE}_5 \downarrow$	$\text{minFDE}_1 \downarrow$	NLL ↓	ECE ↓	RNK ↓
CoverNet[2] [39]	4.92 ± 0.15	2.34 ±0.05	10.94 ± 0.27	3.47 ± 0.06	0.05 ± 0.02	15.55 ± 0.73
GVCL-Det [44]	4.55 ± 0.11	2.26 ± 0.05	9.93 ± 0.39	3.60 ± 0.08	0.18 ± 0.02	14.88 ± 0.94
SNGP_U (Without Rules) [45]	4.53 ± 0.09	2.25 ± 0.04	10.31 ± 0.27	3.23 ± 0.01	<u>0.03</u> ± 0.01	13.25 ± 0.19
SNGP (With Rules)	4.31 ± 0.02	2.17 ± 0.01	9.78 ± 0.05	**3.15** ± 0.01	<u>0.03</u> ± 0.01	11.93 ± 0.11
SHIFT (Ours With Chained Prior)	**4.11** ± 0.05	**2.12** ± 0.01	<u>9.24</u> ± 0.12	<u>3.16</u> ± 0.02	**0.02** ± 0.01	<u>11.85</u> ± 0.22
SHIFT (Ours With Unified Prior)	<u>4.15</u> ± 0.04	<u>2.13</u> ± 0.05	**9.23** ± 0.11	**3.15** ± 0.03	**0.02** ± 0.01	**11.52** ± 0.45

computes the weighted average of the absolute difference between predicted probabilities and observed frequencies across predefined probability bins. A lower ECE indicates better calibration, where the model's predicted probabilities reliably correspond to observed frequencies. **Rank (RNK)**: Measures the rank of the ground truth trajectory among all predicted trajectories when sorted by their predicted probabilities [31]. This metric also measures the calibration of the anchor classification, which can be equally understood as a ranking problem. A lower rank indicates the model assigns higher probabilities to trajectories closer to the ground truth.

C. Baselines and Implementation

We compare SHIFT against CoverNet-based classification baselines from previous work, as these best illustrate the impact of the rule integration and last layer modifications. Therefore, our selected baselines include the conventional CoverNet [39], Deterministic Generalized Variational Continual Learning (GVCL-Det) [44] with CoverNet, an uninformed SNGP-CoverNet model (SNGP_U) [45], and a rule-informed SNGP-CoverNet model (SNGP with Rules) that integrates sets of rules similar to the unified prior setup of SHIFT. The SNGP baselines are uncertainty-aware but do not disentangle uncertainty components, serving as The SNGP baselines are uncertainty-aware but do not disentangle uncertainty components, serving as an ablation to our model. The SNGP model only uses epistemic uncertainty.

Regarding the implementation, we use a ResNet50 backbone and a predefined anchor set of 415 modes for all evaluated models. The input is a rasterized BEV (480x480 pixel) image and color-coded drivability area, lanes, walkways, stop lines and agent trajectories as described by Phan-Minh et al. [39]. We encode the agent trajectories with *1-second* observation window and *6-second* prediction horizon. This design aligns with evidence that short-term observations dominate trajectory dynamics, contributing to 80% of the predictive impact as noted in Schöller et al. [48]. This mitigate tracking error propagation and improve OoD generalization through reduced temporal dependencies [35, 3].

D. Experiments

To ensure a reliable comparison with the baselines, each experiment is conducted over five independent runs and we report the mean and standard deviation of the results. We evaluate SHIFT with a unified prior integrating both sets of traffic rules in the following experiments:

- **Full Data**: We compare SHIFT against all baselines using the full nuScenes train-val-test splits to evaluate general performance. We also compare against a version with a chained prior, integrating all traffic rules sequentially, to investigate the flexibility of the task setup in our first training stage.
- **Reduced Data**: We compare SHIFT against the SNGP baselines on training data subsets, using 50% and 10% of training samples respectively, to assess the data efficiency.
- **Geographic Generalization**: We compare SHIFT with a unified prior against the SNGP baselines on location shifts to assess OoD generalization. These experiments include an *in-distribution (ID)* training/testing in the same location as baseline (`Boston` → `Boston` and `Singapore` → `Singapore`) and an *out-of-distribution (OoD)* training/testing across locations (`Boston` → `Singapore` and `Singapore` → `Boston`).
- **Rule Ablation**: We compare SHIFT against variants that incorporate only one of the rule sets, as well as a version without any rules.

These experiments test data efficiency and robustness to geographic distribution shifts and assess uncertainty calibration within these contexts. We also report qualitative visualization and prediction latency. In Appendix B we highlight results related to the impact of the posterior GP.

V. Results and Discussions

A. Full Dataset

Table I presents the performance comparison on the full dataset. Among the SHIFT configurations, the version with a unified prior achieves the best minFDE_1 and RNK scores, while the chained prior setup excels in minADE_1 and minADE_5. More importantly, both variants of SHIFT significantly outperform baseline methods across key metrics, especially in distribution-aware metrics such as the ECE. These results highlight two properties of SHIFT. First, the task setup in the first training stage is flexible enough to accommodate both chained and unified priors without substantial performance trade-offs. Second, the improved calibration of the heteroscedastic GP layer directly impacts the effectiveness of the regularization-based approach.

[2]We use the result reported in Schlauch et al. [45], where they implemented CoverNet with 1 s of history observation and 6 s prediction horizon along with uncertainty integration.

TABLE II: Impact of Reduced Training Data on Performance Metrics. Bold values indicate the best results.

Model	**minADE$_1$** ↓		**minADE$_5$** ↓		**minFDE$_1$** ↓		**NLL** ↓		**RNK** ↓	
Train Data Used(in %) →	50%	10%	50%	10%	50%	10%	50%	10%	50%	10%
SNGP$_U$ (Without Rules)	4.57±0.05	5.00±0.04	2.26±0.04	2.52±0.03	10.40±0.15	11.36±0.22	3.30±0.02	3.60±0.02	14.62±0.17	25.19±0.30
SNGP (With Rules)	4.41±0.04	4.69±0.07	2.21±0.03	2.32±0.05	9.95±0.09	10.62±0.16	3.23±0.02	3.47±0.02	13.10±0.11	17.80±0.41
SHIFT (Ours Unified Prior)	**4.28**±0.05	**4.60**±0.06	**2.15**±0.04	**2.30**±0.03	**9.60**±0.15	**10.21**±0.14	**3.18**±0.04	**3.45**±0.03	**12.19**±0.5	**17.20**±0.53

TABLE III: Comparison of SHIFT for Geographic Generalization. The best results are highlighted in bold. Both in-distribution (ID) and out-of-distribution (OoD) performance are evaluated to assess the model's ability to generalize to new driving scenarios and to quantify performance degradation when encountering unseen environments.

Region Pair (Trained on → Tested on)	**Model**	**minADE$_1$** ↓	**minADE$_5$** ↓	**minFDE$_1$** ↓	**NLL** ↓	**RNK** ↓
Boston → Boston (ID)	SNGP$_U$ (Without Rules)	4.50± 0.04	2.19± 0.02	10.13± 0.11	3.31± 0.06	13.64± 1.31
	SNGP (With Rules)	4.39± 0.05	2.15± 0.03	9.92± 0.11	3.22± 0.02	12.56± 0.14
	SHIFT (Ours With Unified Prior)	**4.27**± **0.02**	**2.14**± **0.03**	**9.61**± **0.04**	**3.19**± **0.01**	**12.23**± **0.22**
Singapore → Singapore (ID)	SNGP$_U$ (Without Rules)	4.43± 0.07	2.20± 0.06	9.83± 0.15	3.31± 0.06	13.64± 1.31
	SNGP (With Rules)	4.35± 0.04	2.19± 0.02	9.71± 0.14	3.29± 0.01	12.95± 0.27
	SHIFT (Ours With Unified Prior)	**4.28**± **0.03**	**2.16**± **0.06**	**9.45**± **0.08**	**3.23**± **0.04**	**12.46**± **0.51**
Boston → Singapore (OoD)	SNGP$_U$ (Without Rules)	4.82± 0.07	2.60± 0.07	10.95± 0.18	3.52± 0.04	18.39± 0.66
	SNGP (With Rules)	4.65± 0.07	2.48± 0.09	10.57± 0.21	**3.46**± **0.04**	17.95± 0.58
	SHIFT (Ours With Unified Prior)	**4.48**± **0.05**	**2.41**± **0.05**	**10.09**± **0.02**	3.47± 0.04	**16.62**± **1.1**
Singapore → Boston (OoD)	SNGP$_U$ (Without Rules)	5.36± 0.07	2.68± 0.08	12.18± 0.26	3.65± 0.01	21.56± 1.40
	SNGP (With Rules)	5.17± 0.06	2.63± 0.08	11.76± 0.25	3.60± 0.05	19.34± 1.03
	SHIFT (Ours With Unified Prior)	**4.97**± **0.14**	**2.50**± **0.05**	**11.21**± **0.36**	**3.54**± **0.04**	**17.55**± **0.48**

B. Reduced Datasets

Table II presents the model performance with reduced training data (50% and 10% of the original dataset). Even in data-scarce environments, SHIFT consistently outperforms baseline methods across all metrics, demonstrating its robustness in low-data regimes. These results highlight the importance of incorporating traffic rules as prior knowledge in autonomous driving, where data collection is costly, and models must generalize effectively.

C. Geographic Generalization

Geographic generalization results in Table III further validate the effectiveness of SHIFT. In in-distribution (ID) scenarios (Boston → Boston and Singapore → Singapore), our model outperforms the baselines, demonstrating improved prediction accuracy (minADE$_1$, minADE$_5$ and minFDE$_1$) and better-calibrated uncertainty quantification (NLL and RNK). In out-of-distribution (OoD) scenarios (Boston → Singapore and Singapore → Boston), SHIFT demonstrates enhanced robustness, significantly reducing minADE$_1$, minFDE$_1$, and RNK compared to the baselines. The Singapore → Boston transfer exhibits a more pronounced improvement over the baseline, likely due to Singapore's challenging and diverse data providing a stronger generalization capacity. In contrast, the Boston → Singapore transfer shows a more modest gain over SNGP (with Rules), possibly due to limited exposure to Singapore like complex traffic patterns. Nevertheless, the results demonstrate that SHIFT mitigates performance degradation under OoD conditions.

D. Rule Ablation

We analyze the role of traffic rules in the SHIFT model by isolating two rule sets—drivability rules and stop-related rules—as outlined in Sec. III-F. Since stop-related rules (e.g., red light, right-of-way, and pedestrian priority) apply only to a subset of test cases, we group them together in this ablation study to assess their combined impact. The results, shown in Table IV, indicate that stop-related rules play a more significant role in improving top-k displacement metrics, achieving the lowest minADE$_1$ (4.10) and minADE$_5$ (2.13). However, drivability rules, despite having a limited effect, improve calibration by lowering NLL from 3.19 (no rules) to 3.20. This suggests that while drivability constraints may not strongly influence top-k accuracy, they help refine predictive uncertainty. "Only Drivability Rule" refers to global road boundary rules, while "Without Traffic Rules" means neither global nor local rules were used.

Importantly, the unified prior (combining both rule sets) achieves the best overall performance, with improvements in both accuracy (minADE$_5$: 2.13, minFDE$_1$: 9.23) and calibration (NLL: 3.15, RNK: 11.52). This highlights the complementary nature of global constraints (drivability) and situational constraints (stop-related rules) in learning informative priors across all test cases, especially in distribution-aware metrics. Comparing SNGP with SHIFT, we observe that even without rule priors, our model significantly outperforms both the rule-based and rule-free SNGP baselines.

E. Qualitative Results

In Fig. 6 we visually compare predictions of SHIFT and SNGP with rules. At intersections, where agent behavior is

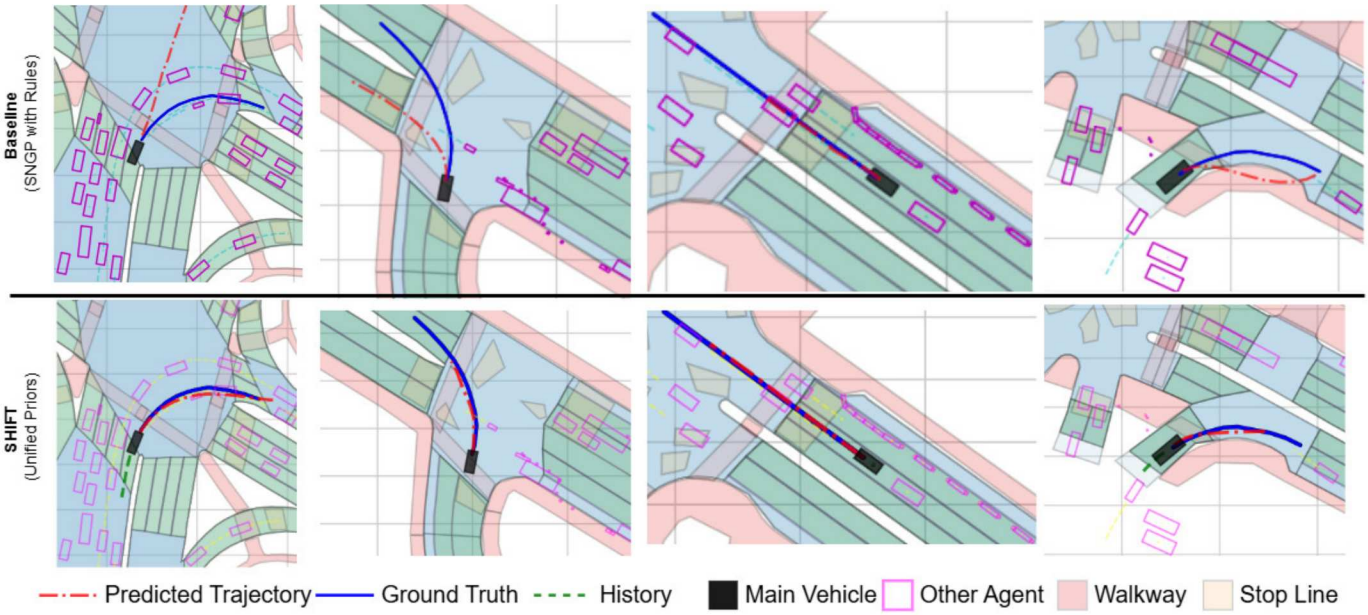

Fig. 6: **Qualitative Comparison of Trajectory Predictions.** The first row displays predictions from our best baseline model, while the second row presents results from SHIFT. Each column represents a different scene. The target agent, for which predictions are made, is highlighted with a black rectangle. The past trajectory (shown only for SHIFT) is also included for reference. Best viewed in color.

TABLE IV: Ablation Study: Traffic Rules Impact on SHIFT.

Configuration	$\text{minADE}_1 \downarrow$	$\text{minADE}_5 \downarrow$	$\text{minFDE}_1 \downarrow$	NLL ↓	RNK ↓
SHIFT (Unified Prior)	4.15	**2.13**	**9.23**	**3.15**	**11.52**
SHIFT (Only Stop Rules)	**4.10**	**2.13**	9.28	3.17	12.10
SHIFT (Only Drivability Rule)	4.25	2.22	9.63	3.20	12.34
SHIFT (Without Traffic Rules)	4.22	2.16	9.57	3.19	12.73
SNGP_U (Without Rules)	4.53	2.25	10.31	3.23	13.25

inherently stochastic, SHIFT more accurately predicts the correct turn direction. Similarly, in scenarios involving stopping areas and multiple nearby vehicles, our model's predictions align more closely with the ground truth trajectory than those of the baseline. Additional visualizations are available in the Appendix E. Notably, in overtaking scenarios, both models performed comparably.

F. Runtime Consideration

Computational efficiency is crucial for autonomous driving applications. Our benchmarks indicate that SHIFT achieves an average prediction latency of $7.4\,\text{ms}$ per sample on a single NVIDIA RTX A5000, compared to $5.6\,\text{ms}$ for CoverNet. This demonstrates that SHIFT incurs only a minimal computational overhead while enhancing trajectory prediction capabilities.

VI. Conclusion

We have introduced SHIFT, an uncertainty-aware trajectory predictor that integrates prior knowledge of traffic rules into deep learning models through a scalable probabilistic formulation. By explicitly modeling disentangled heteroscedastic uncertainty, SHIFT captures both aleatoric and epistemic uncertainties, enabling a more accurate representation of agent behavior while reducing overconfidence. Our empirical evaluations on nuScenes demonstrate that SHIFT produces not only more accurate predictions but also better-calibrated uncertainty estimates, due to the incorporation of rules as both chained and unified priors. The demonstrated flexibility allows end users to incrementally train and adapt the model by integrating new, user-defined driving rules as needed. Such adaptability makes SHIFT particularly well-suited for safety-critical applications in autonomous driving, where agent behaviors are governed by complex social norms and traffic regulations. In future work, we aim to extend this framework to incorporate interactive priors for multi-agent coordination and validate its efficacy within closed-loop planning pipelines.

VII. Limitations

SHIFT demonstrates promising results by integrating traffic rules into trajectory prediction. However, several open research questions remain. First, SHIFT so far been demonstrated only in a trajectory classification context, where compliance is evaluated based on predefined trajectory anchors. Adapting it to other trajectory decoding strategies, such as regression, remains an interesting topic. Second, SHIFT does not guarantee that the integrated traffic rules are actually informative of the behavior of the agents. Measuring the alignment between informative priors and observations could help practitioners in selecting suitable task setups. Third, our rule-filtering approach for generating synthetic training labels is based on static traffic rules. Natural language rules that require temporal evaluations are considerably more difficult to formalize using the RAG-LLM approach. Fourth, prior knowledge about traffic rules must map to observable elements in the model input, such as road geometry and traffic signs. Missing relevant inputs (e.g., occluded traffic signs, unencoded lane types) prevent the

model from learning anything informative from the synthetic training labels. This also applies, for example, to rasterized image inputs, where distinguishing finer details such as solid lines vs dashed lines might be challenging due to inherent limitations in the pixel resolution. Despite these limitations, the modular design of SHIFT reveals an exciting path for future prior knowledge integration, as it can be scaled with the development of both rule formalization and prediction models.

Acknowledgment. This work is partially funded by the German Federal Ministry for Economic Affairs and Climate Action within the project "nxtAIM" and Continental Automotive. We also thank Yue Yao for listening to our ideas and providing valuable feedback from a prediction perspective.

References

[1] Mohammadhossein Bahari, Ismail Nejjar, and Alexandre Alahi. Injecting knowledge in data-driven vehicle trajectory predictors. *Transportation Research Part C: Emerging Technologies*, 128:103010, 2021.

[2] Carlos Barajas, Gianoberto Giampieri, Stefano Sabatini, and Nicola Poerio. Addressing mode collapse in trajectory prediction: A maneuver-oriented metric and approach. In *2024 IEEE Intelligent Vehicles Symposium (IV), Jeju-Island, Korea*, 2024.

[3] Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. RED: A Simple but Effective Baseline Predictor for the TrajNet Benchmark. In *Computer Vision – ECCV 2018 Workshops*, 2018.

[4] Freddy A Boulton, Elena Corina Grigore, and Eric M Wolff. Motion prediction using trajectory sets and self-driving domain knowledge. *arXiv preprint*, arXiv:2006.04767, 2020. URL https://arxiv.org/abs/2006.04767.

[5] Mohamed-Khalil Bouzidi, Yue Yao, Daniel Goehring, and Joerg Reichardt. Learning-aided warmstart of model predictive control in uncertain fast-changing traffic. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan*, 2024.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), virtual*, 2020.

[7] Sergio Casas, Cole Gulino, Simon Suo, and Raquel Urtasun. The importance of prior knowledge in precise multimodal prediction. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA*, 2020.

[8] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the 3rd Annual Conference on Robot Learning (CoRL), Osaka, Japan*, 2019.

[9] Bertrand Charpentier, Chenxiang Zhang, and Stephan Günnemann. Training, architecture, and prior for deterministic uncertainty methods. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

[10] Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. A simple probabilistic method for deep classification under input-dependent label noise. *arXiv preprint*, arXiv:2003.06778, 2020. URL https://arxiv.org/abs/2003.06778.

[11] Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. Correlated input-dependent label noise in large-scale image classification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), virtual*, 2021.

[12] Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Jeff Schneider, David Bradley, and Nemanja Djuric. Deep kinematic models for kinematically feasible vehicle trajectory predictions. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France*, 2020.

[13] Erik A. Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux - effortless bayesian deep learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS), virtual*, 2021.

[14] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366–3385, 2022.

[15] Nachiket Deo and Mohan M. Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint*, arXiv:2001.00735, 2020. URL https://arxiv.org/abs/2001.00735.

[16] Lan Feng, Mohammadhossein Bahari, Kaouther Messaoud Ben Amor, Éloi Zablocki, Matthieu Cord, and Alexandre Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. In *Proceedings of the 18th European Conference in Computer Vision (ECCV), Milan, Italy*, 2024.

[17] Vincent Fortuin, Mark Collier, Florian Wenzel, James Allingham, Jeremiah Liu, Dustin Tran, Balaji Lakshminarayanan, Jesse Berent, Rodolphe Jenatton, and Effrosyni Kokiopoulou. Deep classifiers with label noise modeling and distance awareness. *Transactions on Machine Learning Research (TMLR)*, 2022.

[18] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML), New York City, USA*, 2016.

[19] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks

by enforcing lipschitz continuity. *Machine Learning, Springer*, 110:393–416, 2021.

[20] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision ICCV, Montreal, QC, Canada*, 2021.

[21] Weihang Guo, Zachary K. Kingston, and Lydia E. Kavraki. Castl: Constraints as specifications through llm translation for long-horizon task and motion planning. *arXiv preprint*, abs/2410.22225, 2024. URL https://api.semanticscholar.org/CorpusID:273661618.

[22] Steffen Hagedorn, Marcel Hallgarten, M. Stoll, and Alexandru Condurache. The integration of prediction and planning in deep learning automated driving systems: A review. *IEEE Transactions on Intelligent Vehicles*, pages 1–17, 2023.

[23] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning, Springer*, 110:457–506, 2021.

[24] Masha Itkina and Mykel Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction. In *Proceedings of The 6th Conference on Robot Learning*, Proceedings of Machine Learning Research, 2023.

[25] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016.

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS), virtual*, 2020.

[27] Xiao Li, Guy Rosman, Igor Gilitschenski, Jonathan DeCastro, Cristian-Ioan Vasile, Sertac Karaman, and Daniela Rus. Differentiable logic layer for rule guided trajectory prediction. In *Proceedings of the 2020 Conference on Robot Learning (CORL), virtual*, 2020.

[28] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *Proceedings of the 16th European Conference in Computer Vision (ECCV), Glasgow, UK*, 2020.

[29] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS), virtual*, 2020.

[30] Noel Loo, Siddharth Swaroop, and Richard E Turner. Generalized variational continual learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR), virtual*, 2021.

[31] Rui Luo and Zhixin Zhou. Trustworthy classification through rank-based conformal prediction sets. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2407.04407.

[32] Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada*, 2019.

[33] Kumar Manas, Stefan Zwicklbauer, and Adrian Paschke. CoT-TL: Low-resource temporal knowledge representation of planning instructions using chain-of-thought reasoning. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu-Dhabi, UAE*, 2024.

[34] Kumar Manas, Stefan Zwicklbauer, and Adrian Paschke. TR2MTL: LLM based framework for metric temporal logic formalization of traffic rules. In *Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[35] Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many Observations are Enough? Knowledge Distillation for Trajectory Forecasting. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA*, 2022.

[36] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David J. Weiss, Ben Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *Proceedings of the 10th International Conference on Learning Representations (ICLR), virtual*, 2022.

[37] Brian Paden, Michal Cap, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1 (1):33–55, 2016.

[38] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

[39] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), virtual*, 2020.

[40] Janis Postels, Mattia Segù, Tao Sun, Luca Daniel Sieber, Luc Van Gool, Fisher Yu, and Federico Tombari. On the practicality of deterministic epistemic uncertainty. In *Proceedings of the 39th International Conference on*

Machine Learning (ICML) 2022, 2022.

[41] Ahmad Rahimi and Alexandre Alahi. A multi-loss strategy for vehicle trajectory prediction: Combining off-road, diversity, and directional consistency losses. *arXiv preprint*, arXiv:2411.19747, 2024. URL https://arxiv.org/abs/2411.19747.

[42] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20: Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada*, 2007.

[43] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2005. ISBN 97802622568340.

[44] Christian Schlauch, Christian Wirth, and Nadja Klein. Informed priors for knowledge integration in trajectory prediction. In *Proceedings of the 2023 Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), Turin, Italy*, 2023.

[45] Christian Schlauch, Christian Wirth, and Nadja Klein. Informed spectral normalized Gaussian processes for trajectory prediction. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI) 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS)*, 2024.

[46] Jens Schulz, Constantin Hubmann, Julian Löchner, and Darius Burschka. Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC) 2018, Hawaii, USA*, 2018.

[47] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden*, 2018.

[48] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the Constant Velocity Model Can Teach Us About Pedestrian Motion Prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020.

[49] Florian Seligmann, Philipp Becker, Michael Volpp, and Gerhard Neumann. Beyond deep ensembles: A large-scale evaluation of bayesian deep learning under distribution shift. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA*, 2023.

[50] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA*, 2022.

[51] Ravid Shwartz-Ziv, Micah Goldblum, Hossein Souri, Sanyam Kapoor, Chen Zhu, Yann LeCun, and Andrew Gordon Wilson. Pre-train your loss: Easy bayesian transfer learning with informative priors. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA*, 2022.

[52] Joost R. van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint*, arXiv:2102.11409, 2021. URL https://arxiv.org/abs/2102.11409.

[53] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed Machine Learning – A Taxonomy and Survey of Integrating Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[54] Royden Wagner, Ömer Sahin Tas, Marlon Steiner, Fabian Konstantinidis, Hendrik Königshof, Marvin Klemp, Carlos Fernández, and Christoph Stiller. Scenemotion: From agent-centric embeddings to scene-wide forecasts. *arXiv preprint*, abs/2408.01537, 2024. URL https://arxiv.org/html/2408.01537v3.

[55] Yue Yao, Daniel Goehring, and Joerg Reichardt. An empirical bayes analysis of object trajectory representation models. In *Proceedings of the 26th IEEE International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain*, 2023.

[56] Yue Yao, Shengchao Yan, Daniel Goehring, Wolfram Burgard, and Joerg Reichardt. Improving out-of-distribution generalization of trajectory prediction for autonomous driving via polynomial representations. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, UAE*, 2024.

[57] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

APPENDIX A
COVARIANCE MATRIX $\kappa(h, h)$ AND ESTIMATION OF COVARIANCE IN SHIFT

A. Random Fourier Features (RFF) Approximation

Computing the exact GP covariance matrix $K(h, h)$ scales cubically with the number of data points N, making it computationally infeasible for large datasets. To mitigate this, we employ Random Fourier Features (RFF) to approximate the kernel function, thereby reducing the computational complexity.

1) RFF for RBF Kernel: The RBF kernel is defined as:

$$\kappa(h_i, h_j) = \exp\left(-\frac{\|h_i - h_j\|^2}{2\sigma^2}\right)$$

According to Bochner's theorem [42], any shift-invariant kernel can be expressed as the Fourier transform of a non-negative measure. For the RBF kernel, the corresponding spectral distribution is Gaussian. Thus, the RFF approximation uses samples from this spectral distribution to construct random features.

2) Construction of RFF: Let $W \in \mathbb{R}^{d\times m}$ be a matrix whose entries are drawn from $\mathcal{N}(0, \frac{1}{\sigma^2})$, and $b \in [0, 2\pi]^m$ be a vector of biases uniformly drawn from $[0, 2\pi]$. The RFF approximation maps each input h_i to a random feature vector $\phi(h_i) \in \mathbb{R}^m$ as follows:

$$\phi(h_i) = \sqrt{\frac{2}{m}}\cos(Wh_i + b)$$

3) Kernel Matrix Approximation: Using RFF, the RBF kernel can be approximated by the inner product of the random features:

$$\kappa(h, h) \approx \Phi\Phi^\top$$

where $\Phi \in \mathbb{R}^{N\times m}$ is the matrix of random feature mappings for all inputs, i.e.,

$$\Phi = \begin{bmatrix} \phi(h_1)^\top \\ \phi(h_2)^\top \\ \vdots \\ \phi(h_N)^\top \end{bmatrix}$$

Substituting the expression for $\phi(h_i)$, we get:

$$\begin{aligned}\kappa(h, h) &\approx \Phi\Phi^\top \\ &= \left(\sqrt{\frac{2}{m}}\cos(WH^\top + b\mathbf{1}^\top)\right) \\ &\quad \times \left(\sqrt{\frac{2}{m}}\cos(WH^\top + b\mathbf{1}^\top)\right)^\top\end{aligned}$$

where $H \in \mathbb{R}^{m\times N}$ is the matrix stacking all input embeddings h_i as columns, and $\mathbf{1} \in \mathbb{R}^{1\times N}$ is a vector of ones.

B. Covariance Matrix Computation

The covariance matrix computation involves the following steps:

1) **Random Fourier Feature Mapping:**
   For each input embedding h_i, compute the random feature vector:
   $$\phi(h_i) = \sqrt{\frac{2}{m}}\cos(Wh_i + b)$$
2) **Kernel Matrix Approximation:**
   Approximate the kernel matrix using the inner product of RFF:
   $$\kappa(h, h) \approx \Phi\Phi^\top$$
   where $\Phi = [\phi(h_1), \phi(h_2), \ldots, \phi(h_N)]^\top$.
3) **GP Posterior Covariance Estimation:**
   Using the Laplace approximation, estimate the posterior covariance matrix:
   $$\Sigma_{g_c}^{-1} \approx \mathbb{I} + \sum_{i=1}^{N} p_{i,c}(1 - p_{i,c})\phi(h_i)\phi(h_i)^\top$$

APPENDIX B
ADDITIONAL RESULTS AND DISCUSSION

In our model's posterior estimation, the Gaussian Process (GP) posterior temperature hyperparameter regulates the trade-off between traffic rule-based priors and observed data. To understand its impact on trajectory prediction, we systematically varied the temperature from 5 to 45 and evaluated its influence on key performance metrics, including Negative Log-Likelihood (NLL), Average Displacement Error (ADE_1, ADE_5), Final Displacement Error (FDE_1), and Expected Calibration Error (ECE).

As illustrated in Fig. 7, our analysis shows that while NLL and ADE_5 remain relatively stable across temperature variations, model calibration fluctuates due to the inherently low baseline ECE values. Notably, temperatures outside the range of 15–30 degrade the accuracy of ADE_1 and ADE_5. We identify 15–30 as the optimal temperature range, striking a balance between predictive accuracy and uncertainty calibration.

This study highlights the significant role of the GP posterior temperature in governing SHIFT's ability to integrate prior knowledge with data-driven learning. Practitioners can optimize model performance for specific applications when tuned alongside other hyperparameters, such as the GP $\mathcal{L}_2$ regularizer. Additionally, the nature of the traffic rules incorporated into the prior plays a crucial role, as different rules impact prediction accuracy and calibration in distinct ways.

APPENDIX C
LLM ENABLED TRAFFIC RULES TO PYTHON CODE GENERATION AND LIMITATIONS

To streamline the creation of traffic rule validation functions for trajectory prediction, we utilized a retrieval-augmented

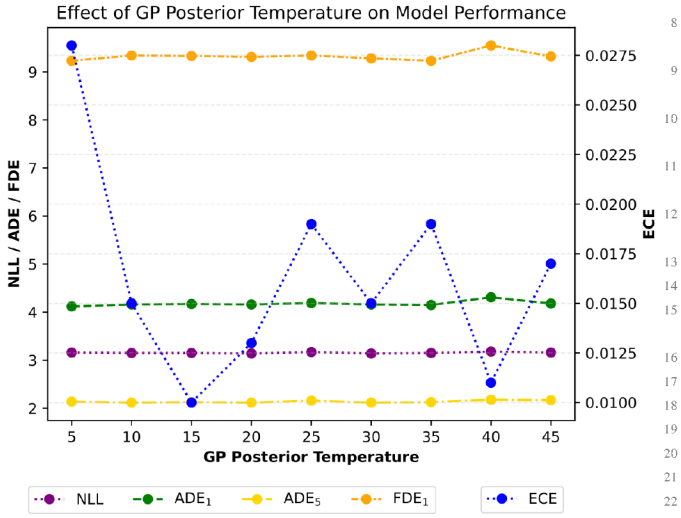

Fig. 7: Impact of GP posterior temperature on trajectory prediction metrics. The x-axis shows temperature values (5-45), with the left y-axis displaying ADE_1, ADE_5, FDE_1, and NLL metrics, and the right y-axis showing ECE. Each metric is uniquely color-coded for visual distinction.

generation (RAG) [26] approach with a large language model (LLM) and prompting as described in Manas et al. [34, 33]. Specifically, the nuScenes map API[3] was employed as a contextual knowledge source. The map API provided access to high-definition (HD) maps, including information on pedestrian crossings, road boundaries, and traffic signs, which served as the basis for generating Python functions tailored to specific traffic rules. By incorporating this map-based context into the LLM prompts, the model generated executable Python code to enforce various constraints in the prior model labeling process.

For instance, when prompted to create a function for traffic rule "`The vehicle trajectory should not cross pedestrian crossings in the presence of pedestrians`" the LLM generated the Python function as shown in Listing 1, which can be directly used in our prior model code base.

Our detailed Prompt file and setup for this is provided in attached code for SHIFT, which includes prompt template and single shot example for this specific task.

```python
from shapely.geometry import LineString

def is_trajectory_crossing_pedestrian_crossings(
    trajectory, translation, rotation, map_name,
    nusc_map, active_crossings=None):
    """
    Check if the trajectory crosses any active
    pedestrian crossings.

    :param trajectory: List of points representing
    the trajectory in local coordinates.
    :param translation: Translation vector to
    convert local coordinates to global coordinates.
    :param rotation: Rotation angle to convert local
     coordinates to global coordinates.
    :param map_name: Name of the map being used (e.g
    ., 'boston-seaport').
    :param nusc_map: NuScenesMap instance for
    accessing map data.
    :return: True if the trajectory crosses any
    active pedestrian crossings, False otherwise.
    """
    # Convert trajectory to global coordinates
    trajectory_global =
    convert_local_coords_to_global(
        trajectory,
        translation,
        rotation,
    )
    trajectory_line = LineString(trajectory_global)

    # Retrieve pedestrian crossings for the map
    if map_name not in nusc_map.ped_crossing:
        ped_crossing_records = nusc_map.ped_crossing
        ped_crossing_polygons = [
            nusc_map.explorer.extract_polygon(record
    ['polygon_token'])
            for record in ped_crossing_records
        ]
    else:
        ped_crossing_polygons = nusc_map.
    ped_crossing[map_name]


    if active_crossings:
        ped_crossing_polygons = [
            nusc_map.explorer.extract_polygon(record
    ['polygon_token'])
            for record in ped_crossing_records
            if record['token'] in active_crossings
        ]

    # Check for intersection with each pedestrian
    crossing
    for crossing_polygon in ped_crossing_polygons:
        if trajectory_line.crosses(crossing_polygon)
    :
            return True  # The trajectory crosses an
     active pedestrian crossing

    return False  # No crossing
```

Listing 1: LLM generated Function to check if trajectory crosses active pedestrian crossings

This function integrates HD map data to determine whether a trajectory crosses active pedestrian crossings, showcasing the capability of the RAG-based LLM to generate high-quality, context-specific Python code. Similarly, other traffic rules defined in Sec. III-F were generated using this approach. While the RAG-based LLM provides substantial benefits in automating code generation, challenges such as hallucinations and incorrect or incomplete context remain prevalent in the process. RAG mitigates these issues to some extent by grounding the generated functions in real-world map sensor data provided by the nuScenes API, enhancing the accuracy and alignment of the outputs with the specified traffic rules. This allowed us to efficiently create prior model labels adhering to a wide range of traffic rules, including both global constraints (e.g., road boundaries) and local constraints (e.g., pedestrian

[3]https://github.com/nutonomy/nuscenes-devkit/tree/master/python-sdk/nuscenes/map_expansion

priority).

Limitations and Mitigation of LLM-based generation. we operate as a *human-in-the-loop* workflow, requiring manual verification and refinement of the LLM-generated outputs. Limitations in the dataset also pose constraints. For instance, the dataset does not include information on emergency vehicles, preventing the generator from producing rules for such scenarios. Furthermore, failures occasionally arise due to the nuanced interpretation of natural language prompts. For example, the generated code used 'shapely.crosses()' or 'shapely.touches()' instead of 'shapely.intersects()' based on the wording of the prompt, which can impact the accuracy of rule-compliant trajectory labels in the prior model. While these issues are not highly detrimental in our soft-prior setup, they underscore the need for careful review to ensure consistency and reliability in the generated outputs. Future work can explore solutions incorporating user-defined constraints and keyword selection or leveraging fine-tuned LLMs with low-rank adaptation. A human-in-the-loop system would significantly streamline the process, making it faster and more efficient by enabling users to verify rules rather than manually creating rule functions from scratch. This approach reduces the complexity of integrating map APIs and codebases for function generation, enhancing usability and scalability.

APPENDIX D
IMPLEMENTATION DETAILS

This section provides key implementation details of our model. For complete configurations and code, refer to our repository.

1. Uncertainty Estimation

To balance epistemic and aleatoric uncertainty, we assign weighting factors of $w_{\text{sngp}} = 0.1$ and $w_{\text{het}} = 0.2$. The Gaussian Process (GP) kernel is approximated using 1,024 inducing points in Random Fourier Features (RFF), ensuring computational efficiency. Spectral normalization with a bound of 2.65 is applied to stabilize training and prevent gradient-related issues.

Temperature Parameters

We introduce temperature parameters to control the influence of prior knowledge and regularization:

- Temperature for GP posterior: 35, regulating influence of traffic rule prior in the GP layer.
- Temperature for feature extractor: 2.8, determining the effective dataset size for L_2 regularization.

These parameters ensure a balanced integration of prior knowledge and empirical observations.

2. Learning Rate Schedule

The learning rate is adjusted based on the training data availability:

- 100% training data: 0.02.
- 50% training data: 0.0005.
- 10% training data: 0.0001.

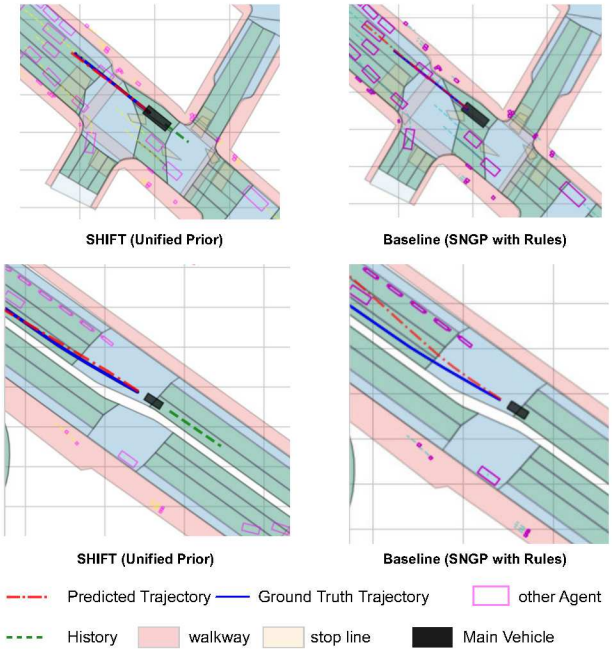

Fig. 8: SHIFT demonstrates better prediction accuracy, closely following the ground truth trajectory in intersection crossing and lane-changing scenarios. In contrast, the baseline model predicts a trajectory with higher speed while crossing the intersection and exhibits lane-change behavior that is closer to another agent, making it potentially unsafe.

A higher learning rate is used for larger datasets to accelerate convergence, while smaller datasets require lower rates to prevent overfitting.

3. Regularization and Early Stopping

To improve generalization, we apply:

- Early stopping with a patience of 15 epochs.
- L_2 regularization with a weight of 0.625 on the extractor output.

4. Parameter Search and Hyperparameter Tuning

We utilize Bayesian optimization with Ray Tune[4] to systematically search for optimal hyperparameters, including the learning rate, spectral norm bound, and feature extractor temperature.

APPENDIX E
ADDITIONAL QUALITATIVE RESULTS

In this section, we present additional qualitative results, highlighting not only the performance of SHIFT but also scenarios where both SNGP (with Rules) and SHIFT perform comparably well, even in complex situations, as well as instances where both methods encounter similar challenges.

[4]https://docs.ray.io/en/latest/tune/index.html

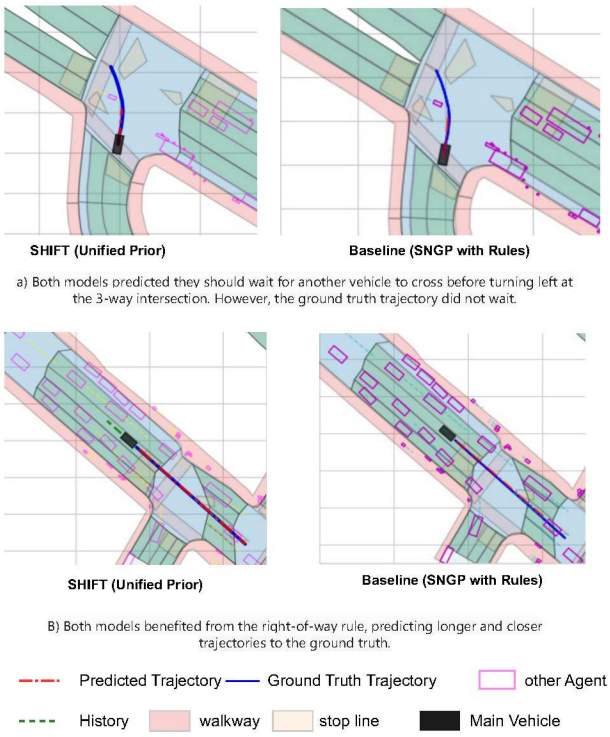

Fig. 9: Both the baseline and SHIFT models perform similarly in these scenarios. In the top image, both models struggle with the decision to either wait for the approaching agent—who has priority—or proceed through the intersection, leading to slightly faster intersection crossing. In the bottom image, both models benefit from right-of-way rules, predicting longer trajectories that are closer to the ground truth.

Cases Where SHIFT Outperforms the Baseline. Figure 8 illustrates a scenario where SHIFT demonstrates superior prediction performance. In both intersection crossing and lane-changing scenarios, SHIFT closely follows the ground truth trajectory, whereas the baseline (SNGP with Rules) predicts a trajectory with higher speed while crossing the intersection. Additionally, during the lane-change maneuver, the baseline's prediction is closer to another agent, which could lead to a potentially unsafe interaction. These results suggest that SHIFT better captures motion dynamics and interactions with surrounding agents, resulting in more reliable predictions.

Cases Where Both Models Perform Similarly. Figure 9 presents scenarios where both SHIFT and the baseline perform on par. In the top image, both models exhibit similar behavior when predicting a left turn at a three-way intersection. While the ground truth trajectory does not wait, both models opt to wait for another agent with priority before proceeding. This indicates that both models incorporate traffic rules effectively but may be overly cautious in certain cases. In the bottom image, both models benefit from right-of-way rules, predicting longer and more accurate trajectories that closely match the ground truth. These results highlight that in structured environments with clear right-of-way constraints, both models can generate reasonable predictions.

Overall, these additional qualitative analyses demonstrate the strengths of SHIFT in improving trajectory prediction while also acknowledging cases where both models exhibit similar behavior and challenges.

APPENDIX F
DATASET STATISTICS AND DETAILS

The **nuScenes dataset** is a large-scale autonomous driving dataset designed for tasks such as 3D object detection, tracking, and trajectory prediction. It provides rich annotations and sensor data, making it suitable for trajectory prediction tasks. Below are the key statistics and details relevant to trajectory prediction:

1. General Dataset Overview

- **Total Scenes**: 1,000 scenes (each 20 seconds long).
- **Total Frames**: ~40,000 keyframes (annotated at 2 Hz).
- **Sensor Data**: Consists of camera, RADAR and LiDAR sensors.
- **Geographic Diversity**: Collected in **Boston** and **Singapore**, covering diverse urban driving scenarios.

The dataset is divided into training, validation, and test splits, with a geographic distribution as follows:

- **Total Data:**
  - Train: 32,186 samples
  - Train-Val: 8,560 samples
  - Validation: 9,041 samples
- **Boston Subset:**
  - Train: 19,629 samples (60.99% of total train sample)
  - Train-Val: 5,855 samples (68.40% of total train-val sample)
  - Validation: 5,138 samples (56.84% of validation sample)
- **Singapore Subset:**
  - Train: 12,557 samples (39.01% of total train sample)
  - Train-Val: 2,705 samples (31.60% of total train-val sample)
  - Validation: 3,903 samples (43.16% of validation sample))

The dataset is geographically diverse, with **Boston** representing North American driving conditions and **Singapore** representing Asian driving conditions. This diversity ensures that models trained on nuScenes generalize well to different regions.

Annotations for Trajectory Prediction

- **Annotated Objects**: 1.4 million 3D bounding boxes across 23 object classes.
- **Relevant Classes for Trajectory Prediction**:
  - Vehicles (car, truck, bus, trailer, etc.).
  - Vulnerable road users such as pedestrians.
- **Trajectory Annotations**:

- Each object has a 3D bounding box annotated at 2 Hz.
- Historical trajectories are available for each object (up to 2 seconds of past data).
- Future trajectories can be extrapolated for prediction tasks.

Trajectory Prediction-Specific Statistics.

- **Trajectory Length**: Future trajectory up to 6 seconds (12 frames at 2 Hz) for evaluation.
- **Interaction Scenarios**:
  - Intersections, roundabouts, and lane changes are common, providing diverse interaction scenarios for trajectory prediction.

Challenges for Trajectory Prediction: The dataset presents challenges such as complex multi-agent interactions (e.g., vehicles and pedestrians), diverse driving scenarios (urban, highway, residential), and partial observations due to occlusions and limited sensor range, making trajectory prediction inherently difficult.