Safe, Out-of-Distribution-Adaptive MPC with Conformalized Neural Network Ensembles

Jose Leopoldo Contreras JCONTR83@STANFORD.EDU
Ola Shorinwa SHORINWA@STANFORD.EDU
Mac Schwager SCHWAGER@STANFORD.EDU
Department of Aeronautics and Astronautics, Stanford University, CA, USA

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

We present SODA-MPC, a Safe, Out-of-Distribution-Adaptive Model Predictive Control algorithm that uses an ensemble of learned models for prediction with a runtime monitor to flag unreliable out-of-distribution (OOD) predictions. When an OOD situation is detected, SODA-MPC triggers a safe fallback control strategy based on reachability, producing a control framework that achieves the high performance of learning-based models while preserving the safety of reachability-based control. We demonstrate the method in the context of an autonomous vehicle, driving among dynamic pedestrians, where SODA-MPC uses a neural network ensemble for pedestrian prediction. We use the maximum singular value of the empirical covariance among the ensemble as the OOD signal for the runtime monitor. We calibrate this signal using conformal prediction to derive an OOD detector with probabilistic guarantees on the false-positive rate, given a user-specified confidence level. During in-distribution operation, the MPC controller avoids collisions with a pedestrian based on the trajectory predicted by the mean of the ensemble. When OOD conditions are detected, the MPC switches to a reachability-based controller to avoid collisions with the reachable set of the pedestrian assuming a maximum pedestrian speed, to guarantee safety under the worst-case actions of the pedestrian. We verify SODA-MPC in extensive autonomous driving simulations in a pedestrian-crossing scenario. Our model ensemble is trained and calibrated with real pedestrian data, showing that our OOD detector obtains the desired accuracy rate within a theoretically-predicted range. We empirically show improved safety and task completion compared with two state-of-the-art MPC methods that also use conformal prediction but without OOD adaptation. Further, we demonstrate the effectiveness of our method with the large-scale multi-agent predictor Trajectron++, using large-scale traffic data from the nuScenes dataset for training and calibration.

Keywords: Conformal prediction, out-of-distribution (OOD) detection, and ensemble learning.

1. Introduction

Robotic autonomy stacks often leverage learning-based models for perception, trajectory prediction, and control. However, in many situations, the performance of these models depends highly on the distribution of the input data encountered at runtime. Learned models can exhibit strong performance when the runtime input data is similarly distributed to the training data (the *in-distribution* setting), but performance suffers when the runtime inputs are significantly different from the training data

[*] This paper has an appendix available at https://arxiv.org/pdf/2406.02436, which provides further empirical results.

(the *out-of-distribution*, or OOD, setting[1]) (Nguyen et al., 2015). This deterioration in performance in OOD settings has limited the utilization of deep-learning models in safety-critical applications. In this work, we introduce a *Safe OOD-Adaptive* Model Predictive Controller (SODA-MPC), which enables robots to operate safely while sharing their task space with other agents that exhibit both in-distribution and OOD behavior.

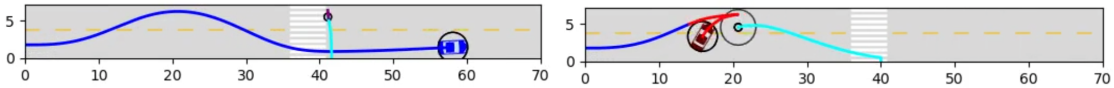

Figure 1: The SODA-MPC algorithm avoids pedestrian collisions in both the nominal (left) and adversarial (right) behavior settings. The algorithm correctly detects the adversarial pedestrian behavior as out-of-distribution, and switches its control strategy to the safe reachable-set-based MPC.

We consider a standard autonomous vehicle (AV) control scheme, which uses an MPC with collision constraints to avoid other agents in the scene. The future trajectories of these agents (required for the collision avoidance constraint in the MPC) are predicted with a learned trajectory prediction model. We address a critical flaw in this architecture: such trajectory predictions are only reliable in in-distribution regimes. Our SODA-MPC algorithm alleviates this problem by introducing a statistically calibrated runtime monitor to detect OOD situations online and switch to a safe fallback controller in those cases. Our runtime monitor uses a neural network ensemble for OOD detection. However, we note that widely-used measures of disagreement for ensemble models are not statistically calibrated signals—they are not tied to an explicit probability of the data being in or out of distribution. Hence, we use conformal prediction to calibrate an OOD detector for the ensemble. When the detector flags an OOD scenario, the MPC ignores the neural network prediction and avoids colliding with the forward reachable sets of the agents, which is a guaranteed-safe fallback controller. The reachable sets are computed assuming a maximum speed bound for the agents.

We demonstrate SODA-MPC in a scenario where an AV must avoid colliding with a pedestrian crossing the road, illustrated in Figure 1. We use a trajectory prediction model ensemble trained from real pedestrian data (Yang et al., 2019). To demonstrate the safety guarantees provided by our controller, we use a handful of held-out trajectories of the original data (which we refer to as *nominal*), and create modified instances of these so that the pedestrian exhibits OOD behavior by running head-on towards the approaching car, a worst-case scenario for the AV. The car must avoid colliding with the pedestrian in both nominal and OOD regimes while attempting to reach a goal position on the road. In our evaluations, we find that our method reaches the goal position in a majority of cases and reliably detects OOD behavior such that it avoids all pedestrian collisions across all tests. We compare against two recent baselines that also use conformal prediction for AV control with learned pedestrian prediction models: SPDE (Lindemann et al., 2023), which calibrates confidence intervals for the learned model, and MARC (Muthali et al., 2023), which calibrates reachable sets for the learned models. Neither method directly addresses OOD behavior. We find that our algorithm is safer than SPDE in OOD regimes and more efficient than MARC which tends to be overly conservative in in-distribution regimes, as demonstrated in Figure 3. In addition, we

[1] We specifically consider the OOD situation known as covariate shift, where the marginal distribution of the input data changes between training and runtime. In our problem, this arises from observed agents expressing unusual behavior not well-represented in the training data.

demonstrate the success of our detector in identifying OOD scenarios when applied to the large-scale trajectory predictor Trajectron++ (Salzmann et al., 2020).

Our contributions are as follows:

- We derive an MPC architecture, SODA-MPC, for learned predictive models that safely adapts to out-of-distribution model predictions. Our MPC switches to a guaranteed-safe reachable set based prediction in out-of-distribution regimes.
- In addition, we introduce an OOD runtime monitor with a guaranteed error rate using neural network ensembles. The monitor is statistically calibrated with conformal prediction, supplying the switching signal for our SODA-MPC.

2. Related Works

2.1. Out-of-Distribution Detection in Neural Networks

Methods for OOD detection can be broadly grouped into training-free and fine-tuning methods. Training-free methods directly leverage the outputs (Liu et al., 2020; Wang et al., 2021; Hendrycks and Gimpel, 2016), e.g., softmax probabilities, and activation patterns (Sun et al., 2021; Dong et al., 2022; Sun and Li, 2022; Sharma et al., 2021) of a trained model to detect OOD inputs. Fine-tuning methods modify the underlying architecture of the model, including the loss function, to estimate the network's confidence in its predictions to distinguish between in-distribution and out-of-distribution settings, e.g., (DeVries and Taylor, 2018; Lee et al., 2017; Hein et al., 2019; Thulasidasan et al., 2019; Madras et al., 2019). We refer interested readers to (Yang et al., 2021) for a more detailed discussion. In the context of robotics, Sinha et al. (2024) detects visual anomalies with an LLM, and enacts a safe fallback MPC if a danger is detected. Although these existing methods work well in many situations, they do not provide provable OOD detection probabilistic guarantees, in general.

2.2. Conformal Prediction in Trajectory Prediction and Motion Planning

Conformal prediction (Vovk et al., 2005; Shafer and Vovk, 2008) provides a black-box, distribution-free method for generating prediction regions from online observations, making it ideal for wide-ranging applications, e.g., in automated decision support systems (Straitouri et al., 2023; Babbar et al., 2022) and in safe trajectory prediction and motion planning (Luo et al., 2022b; Lindemann et al., 2023; Dixit et al., 2023; Lekeufack et al., 2023; Sun et al., 2023). In trajectory prediction, some existing methods leverage conformal prediction for the detection of unsafe situations in early-warning systems (Luo et al., 2022b), verification of autonomous systems (Fan et al., 2020; Dietterich and Hostetler, 2022), and motion planning (Chen et al., 2021; Lindemann et al., 2023; Muthali et al., 2023). In our work, we utilize conformal prediction to provide probabilistic guarantees on the safety of a trajectory prediction module and model predictive controller for autonomous navigation of an agent. Similar to out work, (Lindemann et al., 2023) and (Muthali et al., 2023) apply conformal prediction to learned trajectory prediction models to obtain in-distribution safety guarantees for an MPC. In contrast, we use conformal prediction and a safe fallback controller to obtain safety for both in-distribution and OOD settings.

3. Problem Formulation

We consider an autonomous robot operating in an environment with other agents, where the dynamics model of the robot is given by: $\mathbf{x}_{t+1} = f_e(\mathbf{x}_t, \mathbf{u}_t)$, where $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^{n_e}$ represents the state of the robot at time step t, $\mathcal{X}$ represents its workspace, $\mathbf{x}_0 \in \mathcal{X}$ represents its initial state, $\mathbf{u}_t \in \mathcal{U} \subseteq \mathbb{R}^m$ denotes its control input at t, where $\mathcal{U}$ represents the set of permissible control inputs, and $f_e : \mathbb{R}^{n_e} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_e}$ describes the robot's dynamics, which are presumed known.

The robot seeks to navigate to a goal location optimally without colliding with other agents in its environment. Our method can handle multiple or single agents. We do not assume that these dynamic agents cooperate with the robot, allowing for potential adversarial behaviors. Let the random variable Y denote the joint state of multiple agents, which follows an unknown stationary distribution $\mathcal{D}$ over the states of agents, i.e., $Y \sim \mathcal{D}$, where $Y_t \in \mathbb{R}^{n_a}$ denotes the state of the agents at time t, and n_a is the stacked state dimension of all the agents. We use y when referring to an observation of one of these multi-agent states, with y_t referring to the agents' state at time t. The true dynamics of these agents are unknown, but can be approximated with a learned model $f_a : \mathbb{R}^{n_a} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_a}$, learned from a dataset D where $D := \{y^{(1)}, \ldots, y^{(k)}\}$, and $y^{(i)}$ represents the ith complete observed joint state of the agents in the environment.

Assumption 1 *We have access to k independent realizations y of the distribution $\mathcal{D}$, collected in the dataset $D := \{y^{(1)}, \ldots, y^{(k)}\}$.*

This assumption is not limiting in practice, but is essential for the theoretical guarantees provided by our method. Now, we provide a formal statement of the problem.

Problem 1 (Safe Adaptive Controller) *Given the robot's dynamics and a set of observed in-distribution trajectories of agents $D := \{y^{(1)}, \ldots, y^{(k)}\}$, design an MPC controller that enables an autonomous robot to safely navigate its environment, when operating alongside other agents whose trajectories may be "in-distribution" or "out-of-distribution."*

4. Model Predictive Control

In this work, we utilize a learned ensemble, consisting of n individual multi-layer perceptron (MLP) neural networks for trajectory prediction, given a dataset D. Neural network ensembles enable estimation of the probability distribution of the outputs of the model, which is valuable in preventing overfitting and in confidence estimation (Gustafsson et al., 2020). We note that approaches such as Bayesian neural networks (Jospin et al., 2022) and dropout methods for neural networks (Labach et al., 2019) can be considered as special examples of ensemble learning, providing similar advantages. We use relatively small MLPs as the individual models in the ensemble to demonstrate the effectiveness of our algorithm with simple predictors.

Given the neural network ensemble, we summarize the predictions of the individual models within the ensemble using the mean and the covariance of the predictions, given by:

$$\mathbf{s}_{t+1} = \frac{1}{n}\sum_{i=1}^{n} f_a^i(\mathbf{s}_{t-\ell:t}), \quad \Sigma_{t+1} = \frac{1}{n-1}\sum_{i=1}^{n}(f_a^i(\mathbf{s}_{t-\ell:t}) - \mathbf{s}_{t+1})(f_a^i(\mathbf{s}_{t-\ell:t}) - \mathbf{s}_{t+1})^{\mathsf{T}}, \tag{1}$$

where f_a^i represents the ith model in the ensemble, $\mathbf{s}_{t+1}$ denotes the mean prediction of the state of the environment at time step $t+1$ (with the prediction made at time t), and Σ_{t+1} represents the unbiased estimate of the corresponding covariance matrix for the predictions made by the models in the ensemble at time step t. The input to each network is $\mathbf{s}_{t-\ell:t}$, a concatenation of the (observed or predicted) states of the agents over the preceding ℓ steps, including the current time step. We write $\mathbf{s}_{t+1} = f_a(\mathbf{s}_{t-\ell:t})$ to denote the ensemble mean.

We begin with designing a controller for the nominal case, where the trajectories of the dynamic agents follow the distribution given by $\mathcal{D}$. In this case, we design a model predictive controller leveraging the predictive model f_a to enforce collision-avoidance constraints. The corresponding model predictive control (MPC) problem is given by:

$$\begin{aligned} \textit{MPC I:} \quad & \min_{\mathbf{x}_{1:T}, \mathbf{u}_{0:T-1}} \; J(\mathbf{x}_{1:T}, \mathbf{u}_{0:T-1}) \\ & \quad\text{s.t. } \mathbf{x}_{\tau+1} = f_e(\mathbf{x}_\tau, \mathbf{u}_\tau), \;\; \mathbf{s}_{\tau+1} = f_a(\mathbf{s}_{\tau-l:\tau}), \;\; \forall \tau = 0, \ldots, T-1, \\ & \qquad\; g(\mathbf{x}_\tau, \mathbf{s}_\tau) \leq 0, \;\; \forall \tau = 1, \ldots, T, \end{aligned} \tag{2}$$

where $J : \mathbb{R}^{T \cdot n_e} \times \mathbb{R}^{T \cdot m} \to \mathbb{R}$ denotes the objective function and $\mathbf{x}_{1:T}$ and $\mathbf{u}_{0:T-1}$ denote the concatenation of the optimization variables representing the robot's states and control inputs over the pertinent time steps respectively. We assume we have access to s_0, from which we predict the trajectory $\mathbf{s}_{\tau-l:\tau}$, $\forall \tau$, and $\mathbf{x}_0$, the initial state of the robot. We assume that the MPC problem has a planning horizon of T time steps. We specify constraints enforcing initial conditions, collision avoidance, and other undesirable interactions in $g : \mathbb{R}^{n_e} \times \mathbb{R}^{n_a} \to \mathbb{R}$. The MPC problem is re-solved every H time steps with updated observations.

We note that the resulting MPC-based controller (2) fails to provide safety guarantees, particularly when the dynamic agents exhibit OOD behavior. To address this limitation, we develop an adaptive controller, considering both in-distribution and OOD settings. Our OOD controller leverages the reachable set of the dynamic agents, representing the set of states that the dynamic agents can reach over a specified time duration. We denote the reachable set for the agents in the environment at time step t by $\mathcal{R}_t \subseteq \mathcal{S}$, which we compute from: $\mathcal{R}_{t+1} = \textsc{Reach}(\mathcal{R}_t)$, with $\mathcal{R}_0 = \{\mathbf{s}_0\}$, where $\textsc{Reach} : \mathbb{S} \to \mathbb{S}$ and $\mathbb{S}$ denotes all subsets of $\mathbb{R}^{n_a}$. In computing the reachable set of the dynamic agents, e.g., via velocity-based reachability analysis, we make the following assumption:

Assumption 2 *We assume knowledge of a maximum speed $v_{max} \in \mathbb{R}$ for all agents.*

This assumption is realistic, as we can use reasonable bounds for the top speed of agents in the robot's workspace, such as pedestrians. To provide safety assurances in OOD settings, we consider an MPC problem where the predictive model for the state of the agents in (2) is replaced with a reachable-set-based constraint. The corresponding MPC problem is given by:

$$\begin{aligned} \textit{MPC II:} \quad & \min_{\mathbf{x}_{1:T}, \mathbf{u}_{0:T-1}} \; J(\mathbf{x}_{1:T}, \mathbf{u}_{0:T-1}) \\ & \quad\text{s.t. } \mathbf{x}_{\tau+1} = f_e(\mathbf{x}_\tau, \mathbf{u}_\tau), \;\; \mathcal{R}_{\tau+1} = \textsc{Reach}(\mathcal{R}_\tau), \;\; \forall \tau = 0, \ldots, T-1, \\ & \qquad\; g(\mathbf{x}_\tau, \mathcal{R}_\tau) \leq 0, \;\; \forall \tau = 1, \ldots, T, \end{aligned} \tag{3}$$

where we overload notation, letting $g : \mathbb{R}^{n_e} \times \mathbb{S} \to \mathbb{R}$ denote constraints preventing undesirable interactions (e.g., collisions) defined over subsets of $\mathbb{R}^{n_a}$. Although the *MPC II* controller may be quite conservative, this controller can ensure safety even in situations where agents' behaviors do not follow the distribution $\mathcal{D}$.

To design a safe, OOD-adaptive controller for Problem 1, we compose the MPC-based controller for the nominal setting (*MPC I*) with the reachable-set-based controller (*MPC II*) to enable safe navigation by autonomous robots, necessitating the design of a rule for switching between both controllers. Specifically, an autonomous robot must be able to distinguish between dynamic agents which are acting *nominally* from those agents with *OOD* behavior, concepts which we precisely define later in this work. To address this challenge, we introduce an OOD detector, which leverages conformal prediction to provide probabilistic guarantees. We state the OOD detection problem, prior to discussing our proposed approach to solving this problem:

Problem 2 (OOD detection) *Given the set of observed trajectories $D := \{y^{(1)}, \dots, y^{(k)}\}$ and a failure probability $\delta \in (0, 1)$, identify a signal ρ that can correctly determine, with a probability of $1 - \delta$, that an observed trajectory is sampled from the same probability distribution as that which generated the trajectories in set D.*

5. Out-of-Distribution Detection

We utilize the spectral norm ρ_{t+1} of the empirical covariance matrix Σ_{t+1} (1) as our uncertainty measure to detect OOD samples, with $\rho_{t+1} = \|\Sigma_{t+1}\|_2$, noting that Σ_{t+1} is positive semidefinite and symmetric for all t. However, related measures such as the trace, spectral norm, and Frobenius norm of the covariance matrix and techniques such as (Sharma et al., 2021) and (Luo et al., 2022a) can also be used as the uncertainty measure. We note that the techniques in (Sharma et al., 2021) and (Luo et al., 2022a) require access to the whole training dataset, while we only require a handful of held-out data samples. We leverage conformal prediction to provide provable probabilistic safety guarantees for the safe adaptive controller introduced in this work (see Appendix A in (Contreras et al., 2024) for a brief introduction of conformal prediction).

To define a valid prediction region, we use the spectral norm of the empirical covariance matrix as the nonconformity score: larger values of ρ signify greater degrees of nonconformity. The following remark results from Lemma 1 in (Tibshirani et al., 2019), provided that $\rho^{(i)}$ for $i = 1, \dots, N$, the noncomformity measures of the data points in D_{cal}, are exchangeable, with ρ_{t+1} being the nonconformity score associated with the prediction of the pedestrian position at time step $t + 1$.

Remark 1 *Given a failure probability $\delta \in (0, 1)$, a calibration dataset $D_{\text{cal}} \subset D$, and the predictions $f_a^i(\cdot)$, $\forall i$, we have: $P(\rho_{t+1} \leq C) \geq 1 - \delta$, for $t > 0$, where C represents the prediction region associated with δ.*

Remark 1 states a standard result from Conformal Prediction, which allows strong guarantees despite its simplicity. For example, with 10 calibration data points, we can obtain a test with an exact 10% error rate. To understand this more easily, note that this result considers both the calibration data and the new unseen data points as random variables. In other words, this error rate is obtained exactly when marginalizing over all calibration sets drawn i.i.d. from the data distribution. However, in an engineering application, the calibration dataset is sampled once and fixed, which results in a probability distribution over the obtained error rate for the OOD classifier, which requires a more sophisticated statistical analysis (Shafer and Vovk, 2008; Angelopoulos and Bates, 2021) beyond the scope of this paper. We state the distribution of coverage conditioned on the calibration set in the following theorem.

Theorem 2 *Conditioned on a calibration dataset $D_{\text{cal}} \subset D$, the coverage achieved by conformal prediction follows an analytic distribution given by:*

$$P(\rho_{t+1} \leq C) \sim \text{Beta}(N + 1 - K, K), \;\; \forall t, \tag{4}$$

where N is the size of the D_{cal} dataset and K is the index value of the nonconformity score used to set C when the scores in D_{cal} are placed in nondecreasing order.

Proof The proof is presented in (Vovk, 2012). Hence, we omit the proof here. ■

Theorem 2 indicates that the error rate follows a Beta distribution, which can be used to obtain a confidence that the test holds with a given error rate. E.g., we may calibrate for a 10% error rate, but find from this Beta distribution that we will obtain an error rate less than or equal to 10% with 60% probability and at most a 12% error rate with 90% probability. We note that with more samples in our calibration set, we obtain a tighter Beta distribution, resulting in a greater likelihood that we obtain an error rate close to the calibrated value. This more nuanced analysis is discussed in Appendix C.10 in (Contreras et al., 2024), where we calibrate for a 4% error rate, and obtain a 4.4% error rate for our specific calibration set. At runtime, we detect OOD behavior when the non-conformity score ρ_{t+1} exceeds the calibrated threshold C.

We note that *naive generation* of the calibration dataset would result in the violation of the exchangeability assumption, since trajectories of the dynamic agents collected within the same interaction between the agents and the autonomous robot are not independent. In particular, such trajectories are time-correlated, making them non-exchangeable. To address this issue, we form the calibration dataset D_{cal} by uniformly randomly removing one data point from each trajectory in D_{train} and placing the sampled data in the calibration set. We use the remaining datapoints in the trajectory for training the neural networks. We provide a more detailed discussion of the data-generation procedure in Appendix C.2 in (Contreras et al., 2024). We note that the resulting calibration dataset satisfies the exchangeability assumption, by satisfying the stronger assumption that trajectories are independent and identically distributed. To complete the calibration procedure in the conformal prediction framework, we compute the noncomformity measure associated with each data point in D_{cal} and place them in nondecreasing order, to be used in the specification of C based on the quantile of the empirical distribution corresponding to the desired value of δ.

6. Safe, Out-of-Distribution-Adaptive MPC

Given an OOD detector, we compose a safe OOD-adaptive control architecture that uses the MPC-based controller in (2) with the nominally-acting, dynamic agents, and switches over to a conservative reachable-set-based approach when OOD behavior is detected. Algorithm 1 summarizes our proposed method *SODA-MPC*: Safe, Out-of-distribution-Adaptive Model Predictive Control, which is illustrated in Figure 2 and described in greater detail in Appendix B in (Contreras et al., 2024).

7. Simulations

We evaluate the performance of our controller in both in-distribution and OOD settings, and compare its performance to that of controllers presented in (Muthali et al., 2023) and (Lindemann et al., 2023), which utilize conformal prediction to provide probabilistic safety guarantees, in a problem where an autonomous vehicle attempts to safely navigate to its destination without colliding with a pedestrian crossing the road at a crosswalk. In the in-distribution case, which we refer to as *nominal* behavior, the pedestrian follows a trajectory from the test dataset, D_{test}. Meanwhile, in the OOD case referred to as *insurance fraud* behavior, the pedestrian attempts to force a collision by approaching the vehicle at maximum speed, which constitutes a worst-case scenario, after initially acting nominally for 1.3

seconds. To overcome such an attack, an autonomous vehicle must detect the OOD behavior quickly and adapt its controller to avoid a collision.

Algorithm 1: SODA-MPC: Safe, Out-of-distribution-Adaptive Model Predictive Control

```
Input: Calibraton Dataset $D_{\text{cal}}$ and
       Failure Probability $\delta$ or Index $K$.
Calibrate the OOD Detector.
for $t \leftarrow 0, H, 2H, \dots$ do
    // Observe the environment
    $y_t \leftarrow$ Sensor(t);
    // Detect OOD behavior
    $B \leftarrow$ OOD_Detector($y_t$);
    // Execute the MPC Controller.
    if $B = 0$ then
        // Nominal Controller.
        $(u_t, \dots, u_{t+T-1}) \leftarrow$ MPC I (2)
    else
        // Reachable-Set-Based Controller.
        $(u_t, \dots, u_{t+T-1}) \leftarrow$ MPC II (3)
    Apply $u_{t:t+H}$.
end
```

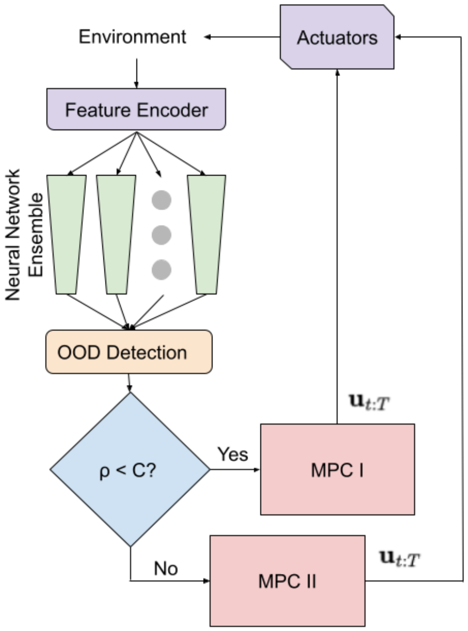

Figure 2: Architecture of SODA-MPC: SODA-MPC detcts OOD behavior from online observations to select a safe control strategy, depending on the behavior of other agents.

7.1. Small, Feedforward MLP Trajectory-Prediction Models

Here, we examine the effectiveness of SODA-MPC in detecting OOD behavior, in problems where the trajectory-prediction model consists of small, feedforward MLPs. Specifically, we predict the future state of a pedestrian using an ensemble of identical MLPs, each with 66 parameters in total, trained on the VCI dataset (Yang et al., 2019). We present the architecture of these models and the associated training dataset in Appendix C in (Contreras et al., 2024).

Empirical Results. We assess the performance of our control algorithm *SODA-MPC* against four baselines, namely: (a) *SPDE* (Lindemann et al., 2023); (b) *MARC* (Muthali et al., 2023); (c) *Reachable Sets Only*, a variant of SODA-MPC utilizing only the MPC II controller; and (d) *Ensembles Only*, a variant of SODA-MPC utilizing only the MPC I controller. We examine each algorithm in 20 simulations, presenting the results achieved by each algorithm in the subsequent discussion. Each simulation terminates in one of three ways: (a) the autonomous vehicle goes past the pedestrian while avoiding collisions, (b) the autonomous vehicle does not go past the pedestrian but avoids a collision (e.g., when stopping to allow the pedestrian to finish crossing the road), or (c) the autonomous vehicle collides with the pedestrian. In SODA-MPC, we utilize a failure probability of approximately 4% in the conformal prediction procedure for computing prediction regions, associated with $C = 0.012$. In contrast, the uncertainty quantification for the MARC algorithm (Muthali et al.,

2023) corresponds to a failure probability of 5% and the ball-shaped confidence areas for the SPDE algorithm (Lindemann et al., 2023) correspond to a failure probability of 5%. In all the figures, we depict the past trajectory of the autonomous vehicle in blue when the autonomous vehicle detects nominal behavior and in red when the autonomous vehicle detects OOD behavior. We depict the past trajectory of the pedestrian in cyan. In Figure 3, we show that SODA-MPC and the more conservative methods MARC and *Reachable Sets Only* do not collide with the pedestrian, even when the pedestrian attempts to force a collision. In the nominal case, the *Ensembles Only* method achieves the highest success rate at 70%, followed jointly by SODA-MPC and SPDE, which achieve a success rate of 60%. Consequently, SODA-MPC provides a desirable tradeoff across both behavior modes.

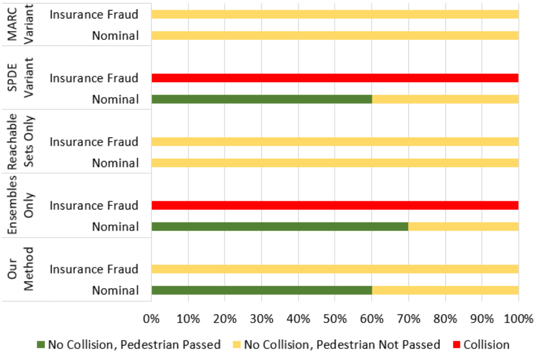

Figure 3: In a pedestrian-crossing scenario, only SODA-MPC (our method) both passed the horizontal position of the pedestrian in the nominal case and avoided collisions with the pedestrian in the *insurance fraud* case, compared to the other baselines.

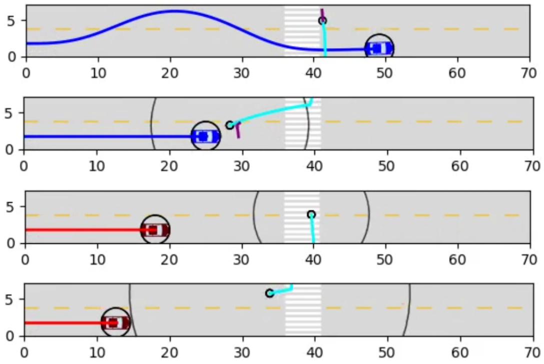

Figure 4: In the top-two rows, the *Ensembles Only* method avoids collisions in the nominal case (top) and collides in the *insurance fraud* setting (bottom). In the bottom-two rows, the overly-conservative *Reachable Sets Only* method avoids collisions in both settings.

While the *Ensembles Only* method could reliably avoid collisions in the nominal case, the method failed to avoid collisions in the *insurance fraud* case, since the predictions of the pedestrian's trajectory differed wildly from the actual trajectory taken by the pedestrian, as depicted in Figure 4. In contrast, the *Reachable Sets Only* method avoids collisions in all settings; however, the autonomous vehicle was consistently unable to reach its desired goal location during the simulation, as illustrated in Figure 4. We note that the MARC algorithm achieved similar conservative results to those of the *Reachable Sets Only* method. Although SPDE remains safe while navigating the vehicle to its goal in the nominal case, SPDE does not avoid collisions when the pedestrian attempts to forcefully collide with the vehicle: this can be attributed to the erroneous predictions generated by the neural network ensemble. Though the method adds a buffer around the position of the pedestrian based on the estimated uncertainty of its prediction, the uncertainty is calibrated under the assumption that the true pedestrian trajectory is drawn from the same distribution as in the training data, while in the *insurance fraud* setting, it is drawn from an altogether different distribution. In contrast, SODA-MPC avoids collisions in all settings, e.g., in the *insurance fraud* setting, the controller switches from the nominal control strategy to the reachable-set-based control strategy after detecting OOD behavior, depicted in Figure 1. We provide empirical results on the statistical guarantees of our OOD detector in Appendix C.10 in (Contreras et al., 2024), showing that our detector achieves a false-positive rate of 4.4% while correctly identifying OOD behavior 91.3% of the time.

7.2. Trajectron++, a Large-Scale Trajectory-Prediction Model

Here, we evaluate the performance of our proposed OOD detector applied to the large-scale trajectory predictor Trajectron++ (Salzmann et al., 2020) as the trajectory predictor on the nuScenes (Caesar et al., 2020) dataset. We provide additional setup details in Appendix C.11 in (Contreras et al., 2024).

Empirical Results. For the calibration procedure, we set $K = 97$ for $N = 100$ for the calibration set point, yielding a detection threshold of $C = 0.933$. Here, predictions are evaluated at the trajectory level, rather than at individual time steps: a trajectory as a whole is considered to be anomalous if the uncertainty value for at least one of its constituent time steps is greater than the detection threshold, and is considered to be nominal otherwise. Nominal cases cover 6230 trajectories of pedestrians in the dataset, and OOD cases use 230 trajectories with shifted pedestrian behavior. We use a calibrated error rate of $\delta \approx 4\%$ for individual data points. In this setting, our OOD detector correctly identifies 5361 (86.1%) of the nominal trajectories as nominal, with a false-positive rate of 13.9% (869 trajectories). Likewise, our OOD detector correctly identifies 211 (91.7%) of the *insurance fraud* trajectories as OOD, with a false-negative rate of 8.3% (19 trajectories), summarized in Table 2 in Appendix C.11 in (Contreras et al., 2024). Compared to an ensemble of identically trained MLP models, the increase in uncertainty seen between different modes when an agent was displaying anomalous behavior was still detectable but less pronounced: we believe this may be ascribed to the cost function of the Trajectron++ model being specifically designed to promote divergence between the prediction modes, rather than training them all to react identically to training data points. This phenomenon was counteracted by evaluating predictions at the trajectory level, which increases the overall sensitivity of the system to detections of anomalous data points. As trajectories displaying *insurance fraud* behavior still have a large number of data points below the detection threshold, this vastly improved the detection rate for anomalous behavior, although the increased sensitivity did increase the incidence of nominal data points being mischaracterized as anomalous compared to the experiments with ensembled feedforward MLP models.

8. Conclusion, Limitations, and Future Work

We introduce SODA-MPC, a safe, adaptive controller that enables autonomous robots to safely navigate in their environments in nominal or OOD settings. SODA-MPC is comprised of a nominal controller based on learned trajectory predictors for in-distribution settings, and safe-by-design controller based on reachable sets for OOD cases. Our OOD detector uses conformal prediction to provide provable probabilistic guarantees on the validity of the prediction regions. We demonstrate the safety and efficacy of our controller in a pedestrian-crossing scenario. With SODA-MPC, the autonomous vehicle avoids collisions with the pedestrian in all settings, and is able to pass the position of the pedestrian when the pedestrian acts in-distribution; whereas some other existing methods either result in collisions, when the pedestrian attempts to force them, or are overly conservative (failing to reach the goal). Our OOD-controller may be too conservative in some settings, especially in dense (congested) scenarios. In future work, we seek to explore the utilization of deep-learned controllers in combination with (or in lieu of) MPC-based controllers in our control framework, which could enable the development of more expressible control frameworks. Further, we are interested in exploring the use of our method in uncertainty quantification to identify and search for areas of the predictor input space not appropriately covered by training data, and in examining other uncertainty quantification metrics for OOD detection.

Acknowledgments

This work was supported in part by ONR grant N00014-23-1-2354, NASA ULI grant 80NSSC20M0163, DARPA grant HR001120C0107, and NSF grant FRR 2342246. Toyota Research Institute provided funds to support this work.

References

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Varun Babbar, Umang Bhatt, and Adrian Weller. On the utility of prediction sets in human-AI teams. *arXiv preprint arXiv:2205.01411*, 2022.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1):149–178, 2023.

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, 2020. URL https://arxiv.org/abs/1903.11027.

Yuxiao Chen, Ugo Rosolia, Chuchu Fan, Aaron Ames, and Richard Murray. Reactive motion planning with probabilisticsafety guarantees. In *Conference on Robot Learning*, pages 1958–1970. PMLR, 2021.

Ford Motor Company, 2018.

Polo Contreras, Ola Shorinwa, and Mac Schwager. Safe, out-of-distribution-adaptive mpc with conformalized neural network ensembles. *arXiv preprint arXiv:2406.02436*, 2024.

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Thomas G Dietterich and Jesse Hostetler. Conformal prediction intervals for markov decision process trajectories. *arXiv preprint arXiv:2206.04860*, 2022.

Anushri Dixit, Lars Lindemann, Skylar X Wei, Matthew Cleaveland, George J Pappas, and Joel W Burdick. Adaptive conformal prediction for motion planning among dynamic agents. In *Learning for Dynamics and Control Conference*, pages 300–314. PMLR, 2023.

Xin Dong, Junfeng Guo, Ang Li, Wei-Te Ting, Cong Liu, and HT Kung. Neural mean discrepancy for efficient out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19217–19227, 2022.

Chuchu Fan, Xin Qin, Yuan Xia, Aditya Zutshi, and Jyotirmoy Deshmukh. Statistical verification of autonomous systems using surrogate models and conformal inference. *arXiv preprint arXiv:2004.00279*, 2020.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319, 2020.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

Alex Labach, Hojjat Salehinejad, and Shahrokh Valaee. Survey of dropout methods for deep neural networks. *arXiv preprint arXiv:1904.13310*, 2019.

Rikard Laxhammar and Göran Falkman. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In *14th international conference on information fusion*, pages 1–8. IEEE, 2011.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

Jordan Lekeufack, Anastasios A Angelopoulos, Andrea Bajcsy, Michael I Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions. *arXiv preprint arXiv:2310.05921*, 2023.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Rachel Luo, Rohan Sinha, Yixiao Sun, Ali Hindy, Shengjia Zhao, Silvio Savarese, Edward Schmerling, and Marco Paovne. Online distribution shift detection via recency prediction. 2022a.

Rachel Luo, Shengjia Zhao, Jonathan Kuck, Boris Ivanovic, Silvio Savarese, Edward Schmerling, and Marco Pavone. Sample-efficient safety assurances using conformal prediction. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 149–169. Springer, 2022b.

David Madras, James Atwood, and Alexander D'Amour. Detecting extrapolation with local ensembles. In *International Conference on Learning Representations*, 2019.

Anish Muthali, Haotian Shen, Sampada Deglurkar, Michael H Lim, Rebecca Roelofs, Aleksandra Faust, and Claire Tomlin. Multi-agent reachability calibration with conformal prediction. *arXiv preprint arXiv:2304.00432*, 2023.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

James Reeds and Lawrence Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific journal of mathematics*, 145(2):367–393, 1990.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Apoorva Sharma, Navid Azizan, and Marco Pavone. Sketching curvature for efficient out-of-distribution detection for deep neural networks. In *Uncertainty in Artificial Intelligence*, pages 1958–1967. PMLR, 2021.

Rohan Sinha, Amine Elhafsi, Christopher Agia, Matthew Foutter, Edward Schmerling, and Marco Pavone. Real-time anomaly detection and reactive planning with large language models. In *Robotics: Science and Systems*, 2024.

Eleni Straitouri, Lequn Wang, Nastaran Okati, and Manuel Gomez Rodriguez. Improving expert predictions with conformal prediction. In *International Conference on Machine Learning*, pages 32633–32653. PMLR, 2023.

Jiankai Sun, Yiqi Jiang, Jianing Qiu, Parth Talpur Nobel, Mykel Kochenderfer, and Mac Schwager. Conformal prediction for uncertainty-aware planning with diffusion dynamics model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, pages 691–708. Springer, 2022.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems*, 34:29074–29087, 2021.

Dongfang Yang, Linhui Li, Keith Redmill, and Ümit Özgüner. Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 899–904. IEEE, 2019.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.