When Environments Shift: Safe Planning with Generative Priors and Robust Conformal Prediction

Kaizer Rahaman AUTOAUTOKAI@GMAIL.COM
Department of Electrical Engineering, Indian Institute of Technology Kharagpur, India

Jyotirmoy V. Deshmukh JDESHMUK@USC.EDU
Thomas Lord Department of Computer Science, University of Southern California, USA

Ashish R. Hota AHOTA@EE.IITKGP.AC.IN
Department of Electrical Engineering, Indian Institute of Technology Kharagpur, India

Lars Lindemann LLINDEMANN@ETHZ.CH
Automatic Control Laboratory, ETH Zürich, Switzerland

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

Autonomous systems operate in environments that may change over time. An example is the control of a self-driving vehicle among pedestrians and human-controlled vehicles whose behavior may change based on factors such as traffic density, road visibility, and social norms. Therefore, the environment encountered during deployment rarely mirrors the environment and data encountered during training – a phenomenon known as distribution shift – which can undermine the safety of autonomous systems. Conformal prediction (CP) has recently been used along with data from the training environment to provide prediction regions that capture the behavior of the environment with a desired probability. When embedded within a model predictive controller (MPC), one can provide probabilistic safety guarantees, but only when the deployment and training environments coincide. Once a distribution shift occurs, these guarantees collapse. We propose a planning framework that is robust under distribution shifts by: (i) assuming that the underlying data distribution of the environment is parameterized by a nuisance parameter, i.e., an observable, interpretable quantity such as traffic density, (ii) training a conditional diffusion model that captures distribution shifts as a function of the nuisance parameter, (iii) observing the nuisance parameter online and generating cheap, synthetic data from the diffusion model for the observed nuisance parameter, and (iv) designing an MPC that embeds CP regions constructed from such synthetic data. Importantly, we account for discrepancies between the underlying data distribution and the diffusion model by using robust CP. Thus, the plans computed using robust CP enjoy probabilistic safety guarantees, in contrast with plans obtained from a single, static set of training data. We empirically demonstrate safety under diverse distribution shifts in the ORCA simulator.

Keywords: Conformal prediction, safe motion planning, distribution shifts, diffusion models

1. Introduction

Autonomous agents increasingly operate in dynamic, open-world environments where deployment-time data can differ from training data—a phenomenon known as *distribution shift*. Such shifts arise from temporal variations, changing environmental conditions, or evolving agent behaviors, and can severely degrade the reliability of learning-based systems. In this paper, we focus on distribution shifts in dynamic environments consisting of uncontrollable agents, where even mild mismatches between training and deployment conditions can compromise safety in domains such as autonomous driving (Filos et al., 2020; Sikar and Garcez, 2024; Arasteh et al., 2025) and robotics (Paudel, 2022).

Motion planning has been explored through both reactive and predictive paradigms (Fox et al., 2002; Mitsch et al., 2013; Dimarogonas et al., 2006; Tanner et al., 2003). Reactive methods respond myopically to an autonomous agent's surrounding but typically lack optimality, whereas predictive methods anticipate future states of uncontrollable agents towards optimality. However, predictions of future states may be unreliable and result in unsafe behavior. In response, *conformal prediction* (CP) was recently used to construct prediction regions that capture uncertainty in predicted future trajectories of uncontrollable agents with a desired probability (Shafer and Vovk, 2008; Cleaveland et al., 2024). Building on this, (Lindemann et al., 2023) introduced *Conformal MPC*, which integrates CP regions within an MPC framework to achieve probabilistic safety. However, previously calibrated CP sets may become invalid under distributions shifts, resulting in safety violations.

Instead, we propose a *robust conformal prediction* framework for predictive motion planning that ensures safety under distribution shifts. Our underlying assumption is that the distribution of the environment is parameterized by a *nuisance parameter*, i.e., an observable, interpretable quantity such as traffic density or road visibility, which provides a structured representation of environmental variations. We then utilize a model of natural variation (MNV) (Robey et al., 2020) that generates inexpensive synthetic data approximating the deployment-time distribution. This enables the construction of CP regions that extend standard CP guarantees to shifted environments and facilitates safe motion planning. Our contributions are summarized as follows: (1) We train a conditional diffusion model that learns the dependence of the nonconformity score – the central element in the study of CP – on the nuisance parameter. (2) We design an MPC that uses robust CP regions that are constructed from cheap, synthetic data generated by the MNV. These robust CP regions account for the discrepancy between the underlying deployment distribution and the diffusion model and guarantee probabilistic safety. (3) We present empirical results in the ORCA simulator to show valid safety coverage under diverse distribution shifts. In comparison, we illustrate that the method from Lindemann et al. (2023) results in more safety violations. We also demonstrate the need for robust CP even when we use synthetic calibration data that is close to the deployment data.

Related Work. Motion planning in dynamic environments has been studied via sampling-based approaches (Kalluraya et al., 2022; Majd et al., 2021; Aoude et al., 2013; Renganathan et al., 2023) and receding-horizon planning (Wei et al., 2022; Wang et al., 2022; Thomas et al., 2021). Model Predictive Control (MPC) provides a receding horizon framework that can incorporate robot dynamics, and can be used for collision avoidance, both in static environments (Zhang et al., 2019, 2020), and in dynamic environments through robust and stochastic MPC formulations (Dixit et al., 2021). Yet, such approaches rely on accurate knowledge of the dynamics and strategies of uncontrollable agents, and safety guarantees may degrade when their behavior changes over time. To mitigate this limitation, recent work has introduced distributionally robust motion planning, in which stochastic collision avoidance constraints are required to hold over an ambiguity set or a family of probability distributions (Hakobyan et al., 2019; Navsalkar and Hota, 2023; Zolanvari and Cherukuri, 2023).

More related to this paper, prior work has embedded CP regions that capture the behavior of uncontrollable agents with a desired probability into the planner by either assuming that the environment does not change (Lindemann et al., 2023; Tonkens et al., 2023; Chen et al., 2021) or by using adaptive CP (Dixit et al., 2023; Yao et al., 2024; Shin et al., 2025; Sheng et al., 2024), which provides average-time guarantees that do not ensure pointwise safety as we do by leveraging robust CP and cheap synthetic data generated by a MNV. CP was also used for reinforcement learning (Yao et al., 2024; Sun et al., 2023), control barrier functions (Zhang et al., 2025; Hsu and Tsukamoto, 2025), and sampling-based search (Sheng et al., 2024), see (Lindemann et al., 2025) for a survey.

2. Problem Formulation

System Description. We consider a discrete-time dynamical system of the form

$$x_{t+1} = f(x_t, u_t), \qquad x_0 = \zeta, \tag{1}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the (ego) agent's state at time t, $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the system dynamics. The ego agent operates in a dynamic environment populated by N uncontrollable agents with states $Y_t = (Y_{t,1}, \dots, Y_{t,N}) \in \mathbb{R}^{Nn}$ whose trajectories $(Y_0, Y_1, \dots)$ are unknown and evolve according to the stochastic process $(Y_0, Y_1, \dots) \sim \mathcal{D}_{\text{test}}$ where $\mathcal{D}_{\text{test}}$ is an (unknown) trajectory distribution encountered during online deployment.

Training Distribution. Let $\mathcal{D}_{\text{train}}$ denote the nominal (offline) distribution from which data can be collected in the form of trajectories to train and calibrate (via conformal prediction) a predictor for the behavior of uncontrollable agents. We refer to a *distribution shift* when $\mathcal{D}_{\text{test}} \neq \mathcal{D}_{\text{train}}$, that is, when trajectory data encountered during deployment deviates from data collected for training. We measure such distribution shifts via a statistical distance $\mathsf{d}(\mathcal{D}_{\text{test}}, \mathcal{D}_{\text{train}})$, where d may denote the Wasserstein distance, KL divergence, or total variation metric. In our setting, distribution shifts arise possibly due to changes in traffic density, road visibility, social norms and couplings between agents. Consequently, safety guarantees for $\mathcal{D}_{\text{train}}$ may no longer hold under $\mathcal{D}_{\text{test}}$.

Idealized Safe Planning Problem. Let $c : \mathbb{R}^n \times \mathbb{R}^{Nn} \to \mathbb{R}$ be a Lipschitz continuous constraint function that indicates safety, e.g., the collision avoidance constraint $c(x_t, Y_t) = \min_j \|x_t - Y_{t,j}\| - \epsilon \geq 0$ imposes that the distance between the states (e.g., positions) of the ego agent and any other (obstacle) agent is at least some safety margin ϵ. We aim to optimize performance while maintaining safety with high probability, leading to a chance-constrained optimization problem:

$$\min_{u_0,\dots,u_{T-1}} \quad J(x,u) \quad \text{s.t.} \;\; x_{t+1} = f(x_t, u_t), \quad t = 0, \dots, T-1, \tag{2a}$$

$$\mathbb{P}_{\mathcal{D}_{\text{test}}}\big(c(x_t, Y_t) \geq 0, \; \forall t \in \{0, \dots, T\}\big) \geq 1 - \delta, \tag{2b}$$

where $J(x,u)$ denotes a user-defined cost (e.g., tracking error or energy consumption), $\delta \in (0,1)$ is a risk tolerance parameter, and $\mathcal{D}_{\text{test}}$ represents the actual (possibly shifted) distribution of the environment during deployment. Solving (2) exactly is generally intractable since $\mathcal{D}_{\text{test}}$ is unknown and may differ from the nominal training distribution $\mathcal{D}_{\text{train}}$. Furthermore, the stochastic process $\mathcal{D}_{\text{test}}$ is high-dimensional and nonlinear, making the chance constraint computationally prohibitive.

We adopt a *data-driven* uncertainty quantification approach using *conformal prediction* (CP). Conceptually, we follow Lindemann et al. (2023) in that we use trajectory predictors for uncontrollable agents Y, construct prediction regions with CP using calibration data from $\mathcal{D}_{\text{train}}$, and then use these to design a model predictive controller for (1). In the absence of distribution shifts, guarantees of the form $\mathbb{P}^{K+1}_{\mathcal{D}_{\text{test}}}\big(c(x_t, Y_t) \geq 0, \; \forall t \in \{0, \dots, T\}\big) \geq 1 - \delta$ can be obtained where K is the number of calibration data.[1] Due to distribution shifts, however, we cannot simply use calibration data from $\mathcal{D}_{\text{train}}$, motivating the need for robust methods that account for distribution shifts.

3. Preliminaries

Vanilla Conformal Prediction. Let $R^{(0)}, \dots, R^{(K)} \sim \mathcal{R}_{\text{test}}$ be $K+1$ i.i.d. random variables, referred to as the nonconformity score. For instance, in a supervised learning setting with inputs

1. Note that the $\mathbb{P}^{K+1}_{\mathcal{D}_{\text{test}}}$ is the $(K+1)$-fold product probability measure of $\mathbb{P}_{\mathcal{D}_{\text{test}}}$ – their relation and the accuracy to which the chance constraint in (2b) is approximated is discussed in more detail in Lindemann et al. (2025).

$X^{(k)}$ and outputs $Z^{(k)}$, a natural choice is $R^{(k)} := \|Z^{(k)} - \mu(X^{(k)})\|$ where μ denotes the predictor; larger values of the nonconformity score indicate poorer predictive performance. The goal of CP is to compute a bound C from the calibration data $R^{(1)}, \ldots, R^{(K)}$ such that the test data $R^{(0)}$ satisfies $\mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}}}(R^{(0)} \leq C) \geq 1 - \delta$, where $1 - \delta$ is a user-specified confidence level. Specifically, $C := \text{Quantile}_{(1+1/K)(1-\delta)}(R^{(1)}, \ldots, R^{(K)})$, i.e., the corrected $(1 - \delta)$-quantile of the empirical distribution over the calibration data, achieves the above probabilistic guarantee (Vovk et al., 2005).

Robust Conformal Prediction. When a distribution shift occurs, standard conformal prediction (CP) coverage guarantees may no longer hold. Let $R^{(1)}, \ldots, R^{(K)} \sim \mathcal{R}_{\text{train}}$ and $R^{(0)} \sim \mathcal{R}_{\text{test}}$ be independent, where the nonconformity scores are induced by $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, respectively. Robust Conformal Prediction (RCP) seeks to ensure valid coverage for any $\mathcal{R}_{\text{test}}$ within an ambiguity set $\mathcal{U}_r = \{\ \mathcal{Q} : \mathsf{d}(\mathcal{Q}, \mathcal{R}_{\text{train}}) \leq r\ \}$ where $r > 0$ bounds the admissible shift magnitude (Cauchois et al., 2024; Aolaritei et al., 2025). The robust coverage constraint can be written as

$$\inf_{\mathcal{R}_{\text{test}} \in \mathcal{U}_r} \mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}}, \mathcal{R}_{\text{train}}}\left(R^{(0)} \leq C^{\text{rob}}\right) \geq 1 - \delta, \tag{3}$$

where C^{rob} denotes a *robust quantile*, often expressed as an inflation of the empirical quantile C as:

$$C^{\text{rob}} = C + \Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)}). \tag{4}$$

Here, $\Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)})$ represents a robustness correction that depends on the ambiguity radius r and, in general, on the nonconformity scores from the calibration data $R^{(1)}, \ldots, R^{(K)}$. The form of $\Delta_{\mathsf{d}}(\cdot)$ depends on how the distributional uncertainty is modeled: (i) in the f-divergence formulation (Cauchois et al., 2024), Δ_{d} is implicitly determined through a convex optimization problem over distributions constrained by an f-divergence ambiguity set; and (ii) in the Lévy–Prokhorov (LP) formulation (Aolaritei et al., 2025), Δ_{d} arises from coupled perturbation radii (ε, ρ) associated with local (infimum-Wasserstein) and global (total-variation) shifts. For clarity of exposition, we use the simplified additive form (4) to represent the effective robustness margin, while acknowledging that, in general, Δ_{d} may depend on multiple factors such as the divergence type, coupled radii (LP metric) and calibration samples. We refer the reader to Appendix 7.2 for a detailed treatment.

Conditional Diffusion Models. We use conditional diffusion models for modeling conditional data distributions (Ho et al., 2020; Han et al., 2022). Let s_0 be a variable that we aim to describe, later corresponding to trajectories Y and nonconformity scores R. The variable s_0 may depend on a contextual parameters $\mathbf{c}$, later corresponding to our nuisance parameter. Conditional diffusion models are distributions $p_\theta(s_0 \mid \mathbf{c})$ which are parameterized by a variable θ that will be learned. Following the classification and regression diffusion (CARD) models (Han et al., 2022), the contextual variable $\mathbf{c}$ is passed through a pre-trained conditional mean encoder $f_\phi(\mathbf{c})$ which estimates $\mathbb{E}[s_0 \mid \mathbf{c}]$ to anchor the diffusion process around a deterministic conditional mean, allowing the generative model to focus on modeling residual uncertainty. These models construct a generative process by progressively corrupting data with Gaussian noise and learning to reverse this process. Full expressions for this forward (noising) process $q(s_j \mid s_0, \mathbf{c})$ and variance schedule β_j are provided in Appendix 7.3, where j denotes the j-th diffusion step. The reverse (denoising) process parametrizes a conditional Gaussian transition:

$$p_\theta(s_{j-1} \mid s_j, \mathbf{c}) = \mathcal{N}(\mu_\theta(s_j, j, f_\phi(\mathbf{c})), \Sigma_j)\,,\ \mu_\theta(s_j, j, f_\phi(\mathbf{c})) = \frac{1}{\sqrt{\alpha_j}}\left(s_j - \frac{\beta_j}{\sqrt{1-\bar{\alpha}_j}}\,\epsilon_\theta(s_j, j, f_\phi(\mathbf{c}))\right), \tag{5}$$

where $\epsilon_\theta(s_j, j, f_\phi(\mathbf{c}))$ predicts the additive noise introduced at diffusion step j. Here, $\alpha_j = 1 - \beta_j$ represents the per-step noise retention factor and $\bar{\alpha}_j = \prod_{k=1}^{j} \alpha_k$ is its cumulative product controlling the effective signal-to-noise ratio across steps, see (Ho et al., 2020) for details. This model thus learns a smooth, context-dependent score field over the contextual parameter $\mathbf{c}$ which is able to approximate underlying, unknown distributions. More details are provided in Appendix 7.3.

4. Safe Planning with Generative Priors and Robust Conformal Prediction

Our central premise is that the uncontrollable trajectories are not governed by a single, fixed distribution $\mathcal{D}_{\text{test}}$, but by a family of distributions indexed by underlying latent factors ζ – termed *natural variations* – that capture phenomena such as environmental changes or shifts in agent behavior that cannot be directly observed but systematically alter the data distribution. These latent factors parameterize a conditional data distribution $\mathcal{D}_{\text{test}}(\zeta)$ that may be encountered during deployment. In practice, however, the latent factor ζ may not be directly observable. We introduce an *observable nuisance parameter* η which is a function of the latent factor ζ, i.e., the nuisance parameter is governed by $\eta(\zeta)$ where we omit dependency on ζ when appropriate. We use η to learn a distribution that serves as a proxy for these latent factors and data-generating distribution. Indeed, we present two variations in Sections 4.1 and 4.2 where we learn trajectory and nonconformity score distributions $\mathcal{D}_{\text{train}}(\eta)$ and $\mathcal{R}_{\text{train}}(\eta)$ to approximate $\mathcal{D}_{\text{test}}(\zeta)$ and $\mathcal{R}_{\text{test}}(\zeta)$ so that, ideally, we obtain $\mathcal{D}_{\text{test}}(\zeta)$=$\mathcal{D}_{\text{train}}(\eta(\zeta))$ and $\mathcal{R}_{\text{test}}(\zeta)$=$\mathcal{R}_{\text{train}}(\eta(\zeta))$. The nuisance parameter acts as a contextual descriptor of the current operating regime, and it may correspond to traffic density, road visibility, and others. Our approach will use this nuisance parameter η to deal with distribution shifts and follow a three step procedure. At each time, we predict the environment, construct robust prediction regions from calibration data sampled from $\mathcal{D}_{\text{train}}(\eta)$ or $\mathcal{R}_{\text{train}}(\eta)$, and solve an MPC iteratively.

4.1. Robust Prediction Regions for Uncontrollable Agents with Generative Trajectory Priors. We now present the construction of prediction regions that contain uncontrollable agents with a probability of at least $1 - \delta$. We use robust conformal prediction with calibration data from the training distribution $\mathcal{D}_{\text{train}}(\eta)$. The distribution $\mathcal{D}_{\text{train}}(\eta)$ here corresponds to a conditional diffusion model $p_\theta(Y \mid \mathbf{c})$ with observable conditioning variables $\mathbf{c} := \eta$. The model $p_\theta(Y \mid \mathbf{c})$ can be trained with data from $\mathcal{D}_{\text{test}}(\zeta)$ for a single or a range of values of ζ. Ideally though, data used for training should have a diverse mix of data from different latent factors ζ. Alternatively, the data used to train $p_\theta(Y \mid \mathbf{c})$ could come from an open access dataset (O'Neill et al., 2024).

Trajectory Prediction. At each timestep t, we use a learned trajectory predictor to produce τ-step-ahead predictions $\hat{Y}_{\tau|t}$ of the environment state Y_τ for all prediction times $\tau \in \{t+1, \ldots, t+H\}$, where H denotes the MPC prediction horizon. The predictor can be any type of predictor and could be trained on the same data that we have trained the training distribution $\mathcal{D}_{\text{train}}(\eta)$ on. As such, we can even train a conditional predictor that provides predictions $\hat{Y}_{\tau|t}(\eta)$ conditioned on η. Our goal now is to compute constants $C^{\text{rob}}_{\tau|t,i} \geq 0$ that define prediction regions of the form

$$\mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}},\mathcal{R}_{\text{train}}}\left(\|\hat{Y}_{\tau|t,i} - Y_{\tau,i}\| \leq C^{\text{rob}}_{\tau|t,i},\ \forall (t,\tau,i) \in \mathcal{S}\right) \geq 1 - \delta \tag{6}$$

where $\mathcal{S} = \{1, \ldots, T-1\} \times \{t+1, \ldots, t+H\} \times \{1, \ldots, N\}$. We note that the prediction region in (6) provides simultaneous coverage over all time steps t, predictions τ, and agents i. Nonconformity scores for simultaneous coverage were proposed in (Cleaveland et al., 2024; Sun and Yu, 2022), but fail to provide coverage under distribution shift. The authors in (Zhao et al., 2024) provided simultaneous coverage under distribution shifts captured by f-divergence measures. Here, we follow

a similar idea, but permit distribution shifts captured by the Lévy–Prokhorov metric while using synthetic data from a conditional generative model to minimize conservatism which is typically induced by robust CP methods – these advantages will later be illustrated in our experiments.

Nonconformity score. We can obtain the guarantees in equation (6) by the nonconformity score

$$R^{(k)} = \max_{(t,\tau,i)\in\mathcal{S}} \frac{\|\hat{Y}^{(k)}_{\tau|t,i} - Y^{(k)}_{\tau,i}\|}{\sigma_{\tau|t,i}} \tag{7}$$

where $Y^{(1)}, \dots, Y^{(K)} \sim \mathcal{D}_{\text{train}}(\eta)$ are K training trajectories, $\hat{Y}^{(1)}_{\tau|t,i}, \dots, \hat{Y}^{(K)}_{\tau|t,i}$ are the corresponding predictions, and $\sigma_{\tau|t,i} > 0$ are normalization constants, see e.g., (Cleaveland et al., 2024; Yu et al., 2026) for their computation. The choice of $\sigma_{\tau|t,i}$ does not affect validity of (6), but well chosen $\sigma_{\tau|t,i}$ result in small prediction regions. This nonconformity score quantifies the discrepancy between the predicted trajectory $\hat{Y}^{(k)}_{\tau|t,i}$ and the true trajectory $Y^{(k)}_{\tau,i}$ for each time step t, prediction τ, and agent i. If there is no distribution shift, the choice $C^{\text{rob}}_{\tau|t,i} := C\sigma_{\tau|t,i}$ ensures that (6) holds.

Distribution shift for the Nonconformity score. The nonconformity scores $\{R^{(i)}\}_{i=1}^{K}$ follow a nonconformity score distribution $\mathcal{R}_{\text{train}}$, i.e., $R^{(1)}, \dots, R^{(K)} \sim \mathcal{R}_{\text{train}}$, which is induced by the distribution $\mathcal{D}_{\text{train}}$.[2] However, we need to consider the nonconformity score distribution $\mathcal{R}_{\text{test}}$ for the nonconformity score $R^{(0)}$ during deployment for which $\mathcal{R}_{\text{train}} \neq \mathcal{R}_{\text{test}}$ may hold. To apply robust conformal prediction as introduced in Section 3, we need to know the statistical distance $\mathsf{d}\big(\mathcal{R}_{\text{test}}, \mathcal{R}_{\text{train}}\big)$, which we can estimate in practice or derive analytical bounds in the case of diffusion models (details provided later). The next result follows directly by applying robust conformal prediction and is a generalization of (Zhao et al., 2024, Lemma 3) by allowing more general statistical distances, such as the Wasserstein and Lévy–Prokhorov metric from (Aolaritei et al., 2025).

Lemma 1 *Let $\delta \in (0,1)$ be a failure probability and $Y^{(0)} \sim \mathcal{D}_{\text{test}}(\zeta)$ and $Y^{(1)}, \dots, Y^{(K)} \sim \mathcal{D}_{\text{train}}(\eta)$ be test and training trajectories for the latent factor ζ and nuisance parameter η. Let $R^{(0)} \sim \mathcal{R}_{\text{test}}$ and $R^{(1)}, \dots, R^{(K)} \sim \mathcal{R}_{\text{train}}$ follow (7) where $\mathcal{R}_{\text{test}}$ and $\mathcal{R}_{\text{train}}$ are the distributions induced by $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{train}}$. If $r \geq 0$ is such that $\mathsf{d}\big(\mathcal{R}_{\text{test}}, \mathcal{R}_{\text{train}}\big) \leq r$, then the choice of $C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i} \cdot (C + \Delta_{\mathsf{d}}(r; R^{(1)}, \dots, R^{(K)}))$ guarantees that the prediction region in (6) holds.*

We refer the reader to Appendix 7.1(a) for the corresponding proof of the lemma. It is easy to see that excessively large bounds $\Delta_{\mathsf{d}}(r; R^{(1)}, \dots, R^{(K)})$ in Lemma 1 can lead to overly conservative prediction regions, which in turn can make the resulting MPC controller (presented in a later section) conservative. For instance, this may be the case when only a fixed distribution $\mathcal{D}_{\text{train}}$ is used, e.g., as in (Zhao et al., 2024). Effectively, this case corresponds to a single nuisance parameter and motivates our approach where we can generate cheap, synthetic data from a conditional diffusion model $\mathcal{D}_{\text{train}}(\eta)$ that can reduce the bound $\Delta_{\mathsf{d}}(r; R^{(1)}, \dots, R^{(K)})$.

4.2. Efficient Prediction Regions with Generative Nonconformity Score Priors. Generating trajectories with diffusion models is computationally expensive and time consuming, potentially prohibitive for real-time application. Instead of generating high-dimensional trajectory data via a learned distribution $\mathcal{D}_{\text{train}}(\eta)$, we here propose to generate nonconformity scores via learning the distribution $\mathcal{R}_{\text{train}}(\eta)$ directly. Indeed, we let $\mathcal{R}_{\text{train}}(\eta)$ be based on a conditional diffusion model $p_\theta(\bar{R} \mid \mathbf{c})$ where now the conditioning variable is $\mathbf{c} = [\,\eta,\, t,\, \tau,\, i\,]$. This is so that $p_\theta(\bar{R} \mid \mathbf{c})$ models

2. We drop the dependency on the latent factor ζ and the nuisance parameter η when clear from the context.

the prediction error $\|\hat{Y}^{(0)}_{\tau|t,i} - Y^{(0)}_{\tau,i}\|$ for trajectories $Y^{(0)} \sim \mathcal{D}_{\text{test}}(\zeta)$. We note that we condition not only on the nuisance parameter η, but also on the current time t, predictions τ, and agents i. This way, we can approximate the nonconformity score in (7) as

$$R^{(k)} = \max_{(t,\tau,i)\in\mathcal{S}} \frac{\bar{R}^{(k)}_{\tau|t,i}}{\sigma_{\tau|t,i}} \tag{8}$$

where $\bar{R}^{(k)}_{\tau|t,i}$ is generated by $p_\theta(\bar{R} \mid \mathbf{c})$, i.e., $\bar{R}^{(k)}_{\tau|t,i} \sim p_\theta(\bar{R} \mid \mathbf{c})$. Training the diffusion model $p_\theta(\bar{R} \mid \mathbf{c})$ follows again the process described in Section 3, with details presented in Appendix 7.3. Data used for training should again have a diverse mix of data with different conditioning variables $\mathbf{c}$ to learn a continuous family of conditional distributions that generalize to unseen nuisance parameters, as we demonstrate in our experiments. The next lemma summarizes our results when the calibration nonconformity scores $R^{(1)}, \dots, R^{(K)}$ follow equation (8) and not equation (7), while the test nonconformity score $R^{(0)}$ again follows (7). Since the test nonconformity score $R^{(0)}$ still follows (7), the proof of the lemma follows almost exactly the same steps and is omitted.

Lemma 2 *Let $\delta \in (0,1)$ be a failure probability and $Y^{(0)} \sim \mathcal{D}_{\text{test}}(\zeta)$ be the test trajectory for the latent factor ζ. Let $R^{(0)} \sim \mathcal{R}_{\text{test}}$ follow (7) where $\mathcal{R}_{\text{test}}$ is the distribution induced by $\mathcal{D}_{\text{test}}$ and $R^{(1)}, \dots, R^{(K)} \sim \mathcal{R}_{\text{train}}$ follow (8) where $\mathcal{R}_{\text{train}}$ is the distribution induced by $p_\theta(R \mid \mathbf{c})$ with $\mathbf{c} = [\eta, t, \tau, i]$. If $r \geq 0$ is such that $\mathsf{d}(\mathcal{R}_{\text{test}}, \mathcal{R}_{\text{train}}) \leq r$, then the choice of $C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i} \cdot (C + \Delta_{\mathsf{d}}(r; R^{(1)}, \dots, R^{(K)}))$ guarantees that the prediction region in (6) holds.*

The nonconformity score in (8), used for calibration, does not depend on the trajectory predictor. The nonconformity score (7), used during deployment, on the other hand does and thereby defines prediction regions with center $\hat{Y}_{\tau|t,i}$ and size $C^{\text{rob}}_{\tau|t,i}$, enabling control design in Section 4.4.

4.3. Estimating Distribution Shifts for Generative Priors. Without any prior knowledge, it is not possible to know the distribution shift r which is ultimately required to apply Lemmas 1 and 2. Note also that r may be different for different contexts $\mathbf{c}$. Generally, r could be treated as a tuning knob that can be empirically adjusted to add robustness margins to the algorithms, e.g., as in robust control (Zhou and Doyle, 1998). We instead aim to estimate the distribution shift r here. For this purpose, we focus on estimating r for Lemma 2, which we also implement in our experiments, while similar methods apply to estimating r for Lemma 1.

Data under context $\mathbf{c}$ available. We will now propose two methods to estimate r when some test data available. In the first method, we assume to have a few trajectories from $\mathcal{D}_{\text{test}}(\zeta)$ – and consequently $\mathcal{R}_{\text{test}}(\zeta)$ – available so that we can estimate r directly, e.g., using (Aolaritei et al., 2025, Algorithm 1) for the LP metric or (Rubenstein et al., 2019) for the f-divergence.

In the second method, we obtain an upper bound for r by quantifying the discrepancy between the true conditional distribution of the prediction error $\|\hat{Y}^{(k)}_{\tau|t,i} - Y^{(k)}_{\tau,i}\|$ for trajectories $Y^{(k)} \sim \mathcal{D}_{\text{test}}(\zeta)$, which we denote by $\bar{\mathcal{R}}_{\text{test}}(\zeta)$, and that generated by our diffusion model $p_\theta(\bar{R} \mid \mathbf{c})$. We employ the analytical 2-Wasserstein bound derived in (Li, 2025, Theorem 1), which states that

$$W_2\big(\bar{\mathcal{R}}_{\text{test}}(\zeta),\, p_\theta(\bar{R} \mid \mathbf{c})\big) \;\leq\; \sum_{j=1}^{T_{\text{diff}}} \beta_j\, M(j)\, \sqrt{H(j)},\;\; M(j) = \exp\Big(\sum_{s\leq j} (L_1(s) + L_2(s)\beta_s)\Big) \tag{9}$$

where $H(j)$ represents the normalized mean square error of the model's predicted diffusion noise, which we estimate on a small held-out validation set $\bar{R}_{\text{test}}(\zeta)$ drawn from $\bar{\mathcal{R}}_{\text{test}}(\zeta)$, L_1, L_2 are drift

regularity constants of the forward process, and T_{diff} is the total number of diffusion steps. We refer the reader to Appendix 7.4 for details. As we obtain an upper bound for $\|\hat{Y}^{(k)}_{\tau|t,i} - Y^{(k)}_{\tau,i}\|$ this way, the choice of $C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i}\Big(C + \Delta_{\mathsf{d}}\Big(\frac{\varepsilon_W}{\sigma_{\min}}; R^{(1)}, \dots, R^{(K)}\Big)\Big)$ with $\sigma_{\min} := \min_{(t,\tau,i)\in\mathcal{S}} \sigma_{\tau|t,i}$ and $\varepsilon_W := \sum_{j=1}^{T_{\text{diff}}} \beta_j M(j) \sqrt{H(j)}$ now makes Lemma 2 valid, see Appendix 7.5 for a proof.

No data under context c available. We may encounter a new context $\mathbf{c}^\star = (\eta^\star, \dots)$ where we have no test data available under the nuisance parameter $\eta^\star$. Assume that we have datasets $R_{\text{test}}(\mathbf{c})$ or $\bar{R}_{\text{test}}(\mathbf{c})$ drawn from $\mathcal{R}_{\text{test}}(\zeta)$ or $\bar{\mathcal{R}}_{\text{test}}(\zeta)$ available, respectively, corresponding to the first or second method presented previously. To handle an "unseen" contexts $\mathbf{c}^\star$, we construct a context-specific test dataset $R_{\text{test}}(\mathbf{c}^\star)$ (or $\bar{R}_{\text{test}}(\mathbf{c}^\star)$) by interpolating between nearby seen contexts. We do so by aggregating data from the I closest seen contexts $\{\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(I)}\}$ to $\mathbf{c}^\star$, where closeness is defined with respect to a chosen metric on the nuisance space. We then construct $R_{\text{test}}(\mathbf{c}^\star) = \cup_{i=1}^{I} R_{\text{test}}(\mathbf{c}^{(i)})$ (or $\bar{R}_{\text{test}}(\mathbf{c}^\star) = \cup_{i=1}^{I} \bar{R}_{\text{test}}(\mathbf{c}^{(i)})$) which serves as the validation dataset for estimating r at the unseen context $\mathbf{c}^\star$ with one of the two previous methods discussed.

4.4. Model Predictive Control with Robust Conformal Prediction Regions. Using these prediction sets, we iteratively solve the following MPC formulation:

$$\min_{u_{t:t+H-1}} \quad J(x,u) \tag{10a}$$

$$\text{s.t. } x_{\tau+1} = f(x_\tau, u_\tau), \quad \tau = t, \dots, t+H-1, \tag{10b}$$

$$\inf_{(Y_0,\dots,Y_t,\dots,Y_{t+H})\in\mathcal{C}_t} c(x_\tau, Y_\tau) \geq 0, \quad \tau = t+1, \dots, t+H, \tag{10c}$$

$$x_\tau \in \mathcal{X},\ u_\tau \in \mathcal{U}, \quad \tau = t, \dots, t+H-1, \tag{10d}$$

where $\mathcal{C}_t := \{Y \,|\, \|\hat{Y}_{\tau|t,i} - Y_{\tau,i}\| \leq C^{\text{rob}}_{\tau|t,i},\ \forall(\tau,i) \in \{t+1,\dots,t+H\} \times \{1,\dots,N\}\}$. With this, we can present the following result whose proof can be found in Appendix 7.1(b).

Theorem 3 *(Closed-loop control.) Given the system in (1), a failure probability $\delta \in (0,1)$, an unknown latent factor ζ, and an observable nuisance parameter $\eta(\zeta)$. Assume that the prediction region in (6) is valid (e.g., via Lemmas 1 or 2 by use of η) and that the optimization problem (10) is feasible at each time $t \in \{0, \dots, T-1\}$, then the closed-loop system satisfies*

$$\mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}},\mathcal{R}_{\text{train}}}\big(c(x_t, Y_t) \geq 0, \forall t \in \{1,\dots,T\}\big) \geq 1-\delta. \tag{11}$$

where $Y \sim \mathcal{D}_{\text{test}}(\zeta)$ is the test trajectory for the latent factor ζ.

5. Case Study

All simulations were executed on a 64-bit `x86_64` workstation equipped with an Intel(R) Core(TM) i7-10700 CPU (2.90 GHz) and 32 GB RAM. The CasADi optimization framework from (Andersson et al., 2019) was employed in an Anaconda-Python environment. In the remainder, we compute efficient prediction regions with the method in Section 4.2 and test and compare four instantiations of the MPC: **Case-0:** This baseline is from Lindemann et al. (2023) which uses the nominal quantile C without robustification computed using calibration data from a single distribution $\mathcal{D}_{\text{train}}$ by fixing a single nuisance parameter. **Case-1:** This baseline uses the quantile C without robustification, but computed using synthetic data from p_θ under the observed η. **Case-2:** This baseline uses the robust quantile $C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i} \cdot (C + \Delta_{\mathsf{d}}(r; R^{(1)}, \dots, R^{(K)}))$ where r is estimated from test data using

(Aolaritei et al., 2025, Algorithm 1) for the ∞-Wasserstein distance. **Case-3:** This baseline uses the robust quantile $C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i} \cdot (C + \Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)}))$ where r is computed analytically using the bound in equation (9). For d, we here use the ∞-Wasserstein distance.

Simulation Setup. We use the ORCA simulator (Van den Berg et al., 2008) configured as a two-lane corridor with one ego agent and two moving obstacles A and B (see Figure 1). The ego starts in the right lane; obstacle A moves oppositely in the same lane, and obstacle B travels in the other lane in the same direction. The ego must safely maneuver (e.g., lane changes) to avoid collisions and reach its goal. Its dynamics follow a kinematic bicycle model discretized via forward Euler with sampling time $\Delta = 1/8$ s (Kong et al., 2015). The ego vehicle has position $p_t := (x_t, y_t)$, orientation θ_t, and velocity v_t. The control inputs are the steering angle ϕ_t and the acceleration a_t.

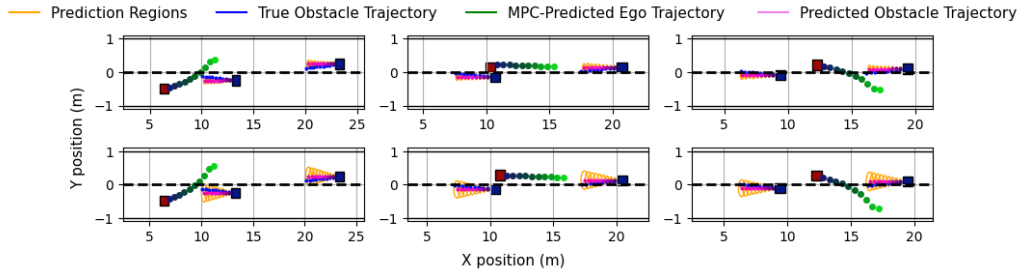

Figure 1: Shown are the ego agent (red), actual (blue) and predicted (pink) trajectories of other agents (blue), along with their prediction regions (yellow). In the top row (case-0), coverage failure leads to collisions. In the bottom row (case-3), the robust prediction regions ensure safety.

Calibration Distribution. The obstacle agents start at $\mathbf{p}_A^{(0)} = [20, -0.5]$ and $\mathbf{p}_B^{(0)} = [30, 0.5]$. Their goals are $\mathbf{g}_i = [x_i^{\text{goal}}, y_i^{\text{goal}}]$, with $x_A^{\text{goal}} = 0$, $x_B^{\text{goal}} = 10$, and $y_i^{\text{goal}} \sim \mathcal{U}[y_i^{\text{nom}} - 0.015,\ y_i^{\text{nom}} + 0.015]$, where $y_A^{\text{nom}} = -0.5$, $y_B^{\text{nom}} = 0.5$. Velocities are sampled as $v_i \sim \mathcal{U}[0.5, 1.0]$ and adjusted by ORCA for feasible obstacle motion. This defines the *training distribution*, from which *test distributions* are obtained by perturbing velocities and goals as described below.

Introducing Distribution Shifts. Each test environment $\mathcal{E}_k$ represents a controlled shift generated by perturbing velocities $v_i \sim \mathcal{U}[0.55, 1.05]$ (common across all test sets) and varying goal latitudes while fixing $x_A^{\text{goal}} = 0$ and $x_B^{\text{goal}} = 10$. Specifically, $\mathcal{E}_k$ is defined by $y_A^{\text{goal}} \sim \mathcal{U}(\mathcal{Y}_A^{(k)})$ and $y_B^{\text{goal}} \sim \mathcal{U}(\mathcal{Y}_B^{(k)})$, where $\mathcal{Y}_A^{(k)} = [-0.06 + 0.03(k-1), -0.03 + 0.03(k-1)]$ and $\mathcal{Y}_B^{(k)} = -\mathcal{Y}_A^{(k)}$, for $k = 1, \ldots, 10$. Increasing k shifts the goal latitudes smoothly outward, yielding a sequence of test environments $\{\mathcal{E}_k\}_{k=1}^{10}$ with progressively larger distributional shifts.

Distribution Splits and Nuisance Parameters. Out of the ten shifted environments $\{\mathcal{E}_k\}_{k=1}^{10}$, the alternate sets $\{\mathcal{E}_2, \mathcal{E}_4, \mathcal{E}_6, \mathcal{E}_8, \mathcal{E}_{10}\}$ are used to train the conditional diffusion model, while the remaining $\{\mathcal{E}_1, \mathcal{E}_3, \mathcal{E}_5, \mathcal{E}_7, \mathcal{E}_9\}$ serve as unseen test distributions. Each environment is characterized by a scalar nuisance parameter η, whose distribution varies across environments with nearly disjoint support. In our MPC setting, a short buffer interval allows estimation of η, defined as the mean absolute lateral velocity over the buffer: $\eta = \frac{1}{2T_b} \sum_{i \in \{A,B\}} \sum_{t=0}^{T_b - 1} \left| \frac{y_i(t+1) - y_i(t)}{\Delta} \right|$, where $y_i(t)$ is the lateral position of obstacle i at time t with sampling interval $\Delta = 1/8$ s. Evaluation across all $\{\mathcal{E}_k\}_{k=1}^{10}$ environments assesses the model's interpolation under intermediate or unseen shifts.

Generalization of the Conditional Diffusion Model. We evaluate the generalization capability of the conditional diffusion model by computing the empirical 2-Wasserstein distance between synthetically generated and test-environment nonconformity score distributions. Synthetic samples

are generated by varying the nuisance parameter η, while each test environment $\mathcal{E}_j$ is defined by a fixed nuisance η_j. As shown in Figure 2, the Wasserstein distance is minimized when $\eta = \eta_j$ and increases as η deviates, indicating increasing distributional mismatch.

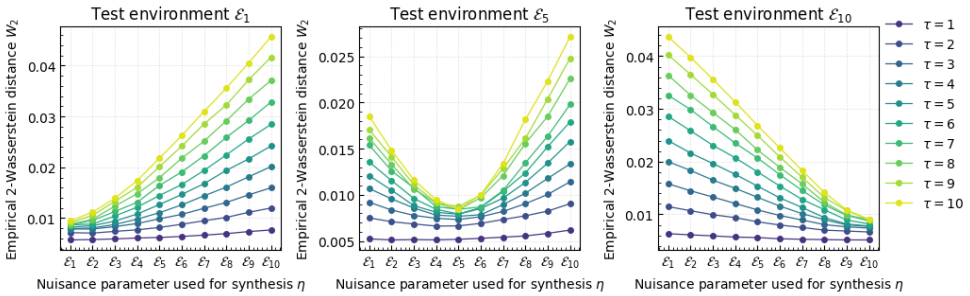

Figure 2: Empirical 2-Wasserstein distance W_2 between synthetically generated trajectory distributions and the true test-environment distribution as a function of the nuisance parameter η used for synthetic generation at time $t = 20$. Each subplot corresponds to a different test environment $\mathcal{E}_j$ characterized by nuisance parameter η_j, and curves within each subplot represent different prediction horizons τ. The Wasserstein distance is minimized when $\eta = \eta_j$ and increases as the nuisance parameter deviates from the test environment, indicating degraded approximation of the test-time nonconformity distribution.

Collision and Coverage Statistics. Figure 3 shows prediction coverage $\mathbb{P}_{\mathcal{R}_{\text{test}}}(\|\hat{Y}_{\tau|t,i} - Y_{\tau,i}\| \leq C^{\text{rob}}_{\tau|t,i},\ \forall (t,\tau,i) \in \mathcal{S})$ and safety coverage $\mathbb{P}_{\mathcal{D}_{\text{test}}}(c(x_t, Y_t) \geq 0,\ \forall t \in \{0,\dots,T\})$ across the ten test environments $\{\mathcal{E}_k\}_{k=1}^{10}$ for both open-loop controller (only solving (10) at time $t = 0$) and closed-loop controller (the MPC solving (10) iteratively). Collisions occur only when the true obstacle trajectories fall significantly outside their predicted regions. In the open-loop setting, where no feedback correction is applied, **Case-0** shows frequent safety violations due to fixed calibration data, while **Case-1** exhibits violations despite using only synthetic data without robustification. Feedback present in the closed-loop controller mitigates these failures, reducing collisions even when coverage deteriorates. Both **Case-2** and **Case-3** maintain near-perfect safety and the target $1 - \delta = 0.9$ coverage across all environments, demonstrating that robust prediction regions effectively ensure safe operation under distributional shifts without being overly conservative, see Figure 1.

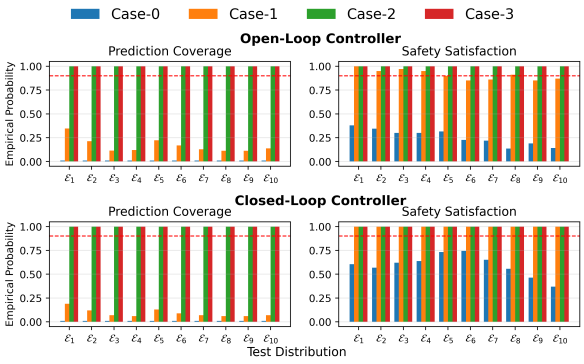

Figure 3: Comparison of prediction and safety coverage across test environments for Cases 0-3 under open- and closed-loop controllers. The red dashed line show $1-\delta = 0.9$.

Time Complexity. We report the time complexity of our framework in Appendix 7.6.

6. Conclusion

We propose a planning framework that maintains probabilistic safety under distribution shifts using robust conformal prediction and conditional diffusion models. Conditioning on an interpretable nuisance parameter, our model generates synthetic data reflecting test-time variations thus enabling safe MPC. Experiments in the ORCA simulator show reliable safety across seen and unseen environments, with future work targeting interactive and non-stationary distribution shifts.

References

Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. Casadi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

Liviu Aolaritei, Zheyu Oliver Wang, Julie Zhu, Michael I Jordan, and Youssef Marzouk. Conformal Prediction under Levy-Prokhorov Distribution Shifts: Robustness to Local and Global Perturbations. *arXiv preprint arXiv:2502.14105*, 2025.

Georges S Aoude, Brandon D Luders, Joshua M Joseph, Nicholas Roy, and Jonathan P How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, 35(1):51–76, 2013.

Fazel Arasteh, Mohammed Elmahgiubi, Behzad Khamidehi, Hamidreza Mirkhani, Weize Zhang, Cao Tongtong, and Kasra Rezaee. Validity Learning on Failures: Mitigating the Distribution Shift in Autonomous Vehicle Planning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15680–15686. IEEE, 2025.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John C Duchi. Robust Validation: Confident Predictions Even When Distributions Shift. *Journal of the American Statistical Association*, 119 (548):3033–3044, 2024.

Yuxiao Chen, Ugo Rosolia, Chuchu Fan, Aaron Ames, and Richard Murray. Reactive Motion Planning with Probabilistic Safety Guarantees. In *Conference on Robot Learning*, pages 1958–1970. PMLR, 2021.

Matthew Cleaveland, Insup Lee, George J Pappas, and Lars Lindemann. Conformal Prediction Regions for Time Series using Linear Complementarity Programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20984–20992, 2024.

Dimos V Dimarogonas, Savvas G Loizou, Kostas J Kyriakopoulos, and Michael M Zavlanos. A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Nonholonomic Non-Point Agents. *Automatica*, 42(2):229–243, 2006.

Anushri Dixit, Mohamadreza Ahmadi, and Joel W Burdick. Risk-Sensitive Motion Planning using Entropic Value-at-Risk. In *2021 European Control Conference (ECC)*, pages 1726–1732. IEEE, 2021.

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.

Angelos Filos, Panagiotis Tigkas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can Autonomous Vehicles Identify, Recover From, and Adapt to Distribution Shifts?. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020.

Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 2002.

Astghik Hakobyan, Gyeong Chan Kim, and Insoon Yang. Risk-Aware Motion Planning and Control Using CVaR-Constrained Optimization. *IEEE Robotics and Automation Letters*, 4(4):3924–3931, 2019.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. CARD: Classification and Regression Diffusion Models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ting-Wei Hsu and Hiroyasu Tsukamoto. Statistical Guarantees in Data-Driven Nonlinear Control: Conformal Robustness for Stability and Safety. *IEEE Control Systems Letters*, 2025.

Samarth Kalluraya, George J Pappas, and Yiannis Kantaros. Multi-robot Mission Planning in Dynamic Semantic Environments. *arXiv preprint arXiv:2209.06323*, 2022.

Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099. IEEE, 2015.

Mengze Li. Non-asymptotic convergence bound of conditional diffusion models. *arXiv preprint arXiv:2508.10944*, 2025.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe Planning in Dynamic Environments using Conformal Prediction. *IEEE Robotics and Automation Letters*, 8(8): 5116–5123, 2023.

Lars Lindemann, Yiqi Zhao, Xinyi Yu, George J Pappas, and Jyotirmoy V Deshmukh. Formal Verification and Control With Conformal Prediction: Practical Safety Guarantees For Autonomous Systems. *IEEE Control Systems*, 45(6):72–122, 2025.

Keyvan Majd, Shakiba Yaghoubi, Tomoya Yamaguchi, Bardh Hoxha, Danil Prokhorov, and Georgios Fainekos. Safe Navigation in Human Occupied Environments Using Sampling and Control Barrier Functions. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5794–5800. IEEE, 2021.

Stefan Mitsch, Khalil Ghorbal, and André Platzer. On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles. In *Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24-June 28, 2013*, 2013.

Atharva Navsalkar and Ashish R Hota. Data-Driven Risk-sensitive Model Predictive Control for Safe Navigation in Multi-Robot Systems. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1442–1448. IEEE, 2023.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

Abhishek Paudel. Learning for Robot Decision Making under Distribution Shift: A Survey. *arXiv preprint arXiv:2203.07558*, 2022.

Venkatraman Renganathan, Sleiman Safaoui, Aadi Kothari, Benjamin Gravell, Iman Shames, and Tyler Summers. Risk Bounded Nonlinear Robot Motion Planning With Integrated Perception Control. *Artificial Intelligence*, 314:103812, 2023.

Alexander Robey, Hamed Hassani, and George J Pappas. Model-Based Robust Deep Learning: Generalizing to Natural, Out-of-Distribution Data. *arXiv preprint arXiv:2005.10247*, 2020.

Paul Rubenstein, Olivier Bousquet, Josip Djolonga, Carlos Riquelme, and Ilya O Tolstikhin. Practical and Consistent Estimation of f-Divergences. *Advances in Neural Information Processing Systems*, 32, 2019.

Glenn Shafer and Vladimir Vovk. A Tutorial on Conformal Prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Shili Sheng, Pian Yu, David Parker, Marta Kwiatkowska, and Lu Feng. Safe POMDP Online Planning among Dynamic Agents via Adaptive Conformal Prediction. *IEEE Robotics and Automation Letters*, 2024.

Jaeuk Shin, Jungjin Lee, and Insoon Yang. Egocentric Conformal Prediction for Safe and Efficient Navigation in Dynamic Cluttered Environments. *arXiv preprint arXiv:2504.00447*, 2025.

Daniel Sikar and Artur Garcez. Evaluation of autonomous systems under data distribution shifts. *arXiv preprint arXiv:2406.20046*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. *arXiv preprint arXiv:2011.13456*, 2020.

Jiankai Sun, Yiqi Jiang, Jianing Qiu, Parth Nobel, Mykel J Kochenderfer, and Mac Schwager. Conformal Prediction for Uncertainty-Aware Planning with Diffusion Dynamics Model. *Advances in Neural Information Processing Systems*, 36:80324–80337, 2023.

Sophia Sun and Rose Yu. Copula Conformal Prediction for Multi-step Time Series Forecasting. *arXiv preprint arXiv:2212.03281*, 2022.

Herbert G Tanner, Savvas G Loizou, and Kostas J Kyriakopoulos. Nonholonomic Navigation and Control of Cooperating Mobile Manipulators. *IEEE Transactions on Robotics and Automation*, 19(1):53–64, 2003.

Antony Thomas, Fulvio Mastrogiovanni, and Marco Baglietto. Probabilistic Collision Constraint for Motion Planning in Dynamic Environments. In *International Conference on Intelligent Autonomous Systems*, pages 141–154. Springer, 2021.

Sander Tonkens, Sophia Sun, Rose Yu, and Sylvia Herbert. Scalable Safe Long-Horizon Planning in Dynamic Environments Leveraging Conformal Prediction and Temporal Correlations. In *Long-Term Human Motion Prediction Workshop, International Conference on Robotics and Automation*, 2023.

Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2008.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Allan Wang, Christoforos Mavrogiannis, and Aaron Steinfeld. Group-based Motion Prediction for Navigation in Crowded Environments. In *Conference on Robot Learning*, pages 871–882. PMLR, 2022.

Skylar X Wei, Anushri Dixit, Shashank Tomar, and Joel W Burdick. Moving Obstacle Avoidance: A Data-Driven Risk-Aware Approach. *IEEE Control Systems Letters*, 7:289–294, 2022.

Jianpeng Yao, Xiaopan Zhang, Yu Xia, Zejin Wang, Amit K Roy-Chowdhury, and Jiachen Li. SoNIC: Safe Social Navigation with Adaptive Conformal Inference and Constrained Reinforcement Learning. *arXiv preprint arXiv:2407.17460*, 2024.

Xinyi Yu, Yiqi Zhao, Xiang Yin, and Lars Lindemann. Signal Temporal Logic Control Synthesis among Uncontrollable Dynamic Agents with Conformal Prediction. *Automatica*, 183:112616, 2026.

Chaoyong Zhang, Duanfeng Chu, Shidong Liu, Zejian Deng, Chaozhong Wu, and Xiaocong Su. Trajectory Planning and Tracking for Autonomous Vehicle Based on State Lattice and Model Predictive Control. *IEEE Intelligent Transportation Systems Magazine*, 11(2):29–40, 2019.

Junhui Zhang, Bardh Hoxha, Georgios Fainekos, and Dimitra Panagou. Conformal Prediction in the Loop: Risk-Aware Control Barrier Functions for Stochastic Systems with Data-Driven State Estimators. *IEEE Control Systems Letters*, 2025.

Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-Based Collision Avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2020.

Yiqi Zhao, Bardh Hoxha, Georgios Fainekos, Jyotirmoy V Deshmukh, and Lars Lindemann. Robust Conformal Prediction for STL Runtime Verification under Distribution Shift. In *2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*, pages 169–179. IEEE, 2024.

Kemin Zhou and John Comstock Doyle. *Essentials of Robust Control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

Alireza Zolanvari and Ashish Cherukuri. Wasserstein Distributionally Robust Risk-Constrained Iterative MPC for Motion Planning: Computationally Efficient Approximations. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 2022–2029. IEEE, 2023.

7. Appendix

7.1. Proofs

(a) Proof of Lemma 1 from Section 4.1

Since $\mathsf{d}\big(\mathcal{R}_{\text{test}}, \mathcal{R}_{\text{train}}\big) \leq r$ holds, robust conformal prediction guarantees via (3) that

$$\mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}},\mathcal{R}_{\text{train}}}\big(R^{(0)} \leq C + \Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)})\big) \geq 1 - \delta. \tag{12}$$

Plugging $R^{(0)}$ from equation (7) into (12) gives us that

$$\mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}},\mathcal{R}_{\text{train}}}\Big(\max_{(t,\tau,i)\in\mathcal{S}} \frac{\|\hat{Y}^{(0)}_{\tau|t,i} - Y^{(0)}_{\tau,i}\|}{\sigma_{\tau|t,i}} \leq C + \Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)})\Big) \geq 1 - \delta \tag{13}$$

$$\Leftrightarrow \quad \mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}},\mathcal{R}_{\text{train}}}\Big(\|\hat{Y}^{(0)}_{\tau|t,i} - Y^{(0)}_{\tau,i}\| \leq \sigma_{\tau|t,i}(C + \Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)})),\ \forall (t,\tau,i) \in \mathcal{S}\Big) \geq 1 - \delta \tag{14}$$

from which equation (6) follows by the choice of $C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i} \cdot (C + \Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)}))$.

(b) Proof of Theorem 3 from Section 4.4

The guarantee in (11) directly follows since the optimization problem (10) is feasible at each time $t \in \{0, \ldots, T-1\}$ and since the constraint (10c) uses the set $\mathcal{C}_t$ that is such that $\mathbb{P}^{K+1}_{\mathcal{R}_{\text{test}},\mathcal{R}_{\text{train}}}\big(Y \in \mathcal{C}_t, \forall t \in \{1, \ldots, T\}\big) \geq 1 - \delta$, which is guaranteed since the prediction regions in (6) are valid.

7.2. Robust Conformal Prediction and Computation of Δ_{d}

This section provides details on the computation and interpretation of the robustness correction term $\Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)})$ introduced in equation (4). We distinguish between two principal formulations in the literature: the *f-divergence ambiguity* approach (Cauchois et al., 2024) and the *Lévy–Prokhorov (LP)* ambiguity approach (Aolaritei et al., 2025).

(a) f-Divergence–Based Robust Conformal Prediction

The f-divergence ambiguity formulation Cauchois et al. (2024) defines the admissible uncertainty set around the training score distribution as

$$\mathcal{U}^f_r(\mathcal{R}_{\text{train}}) = \{\, \mathcal{Q} : D_f(\mathcal{Q} \,\|\, \mathcal{R}_{\text{train}}) \leq r \,\},$$

where $D_f(\cdot\|\cdot)$ denotes an f-divergence and $r > 0$ bounds the shift magnitude. Let

$$R^{(1)}, \ldots, R^{(K)} \overset{\text{i.i.d.}}{\sim} \mathcal{R}_{\text{train}}, \qquad R^{(0)} \sim \mathcal{R}_{\text{test}}.$$

Then, calibration scores from $\mathcal{R}_{\text{train}}$ can be used to construct a robust quantile that guarantees coverage under all $\mathcal{R}_{\text{test}} \in \mathcal{U}^f_r$.

Lemma 4 (Adapted from Cauchois et al. (2024)) *Let $R^{(0)}, \ldots, R^{(K)}$ be independent random variables with $R^{(0)} \sim \mathcal{R}_{\text{test}}$ and $R^{(1)}, \ldots, R^{(K)} \sim \mathcal{R}_{\text{train}}$, satisfying $\mathcal{R}_{\text{test}} \in \mathcal{U}_r^f(\mathcal{R}_{\text{train}})$. For a failure probability $\delta \in (0,1)$, the guarantee $\mathbb{P}\big(R^{(0)} \le C^{\text{rob}}\big) \ge 1-\delta$ holds for*

$$C^{\text{rob}} := \text{Quantile}_{1-\tilde{\delta}}\big(R^{(1)}, \ldots, R^{(K)}\big),$$

where the adjusted level $\tilde{\delta}$ is obtained via one-dimensional convex programs:

$$\delta_K := 1 - g\Big((1+\tfrac{1}{K})\, g^{-1}(1-\delta)\Big),$$
$$\tilde{\delta} := 1 - g^{-1}(1-\delta_K),$$

where the auxiliary function $g : [0,1] \to [0,1]$ is defined as

$$g(\beta) := \inf\left\{ z \in [0,1] \;\middle|\; \beta f\Big(\frac{z}{\beta}\Big) + (1-\beta) f\Big(\frac{1-z}{1-\beta}\Big) \le r \right\},$$

and its pseudo-inverse $g^{-1}(\tau) := \sup\{\beta \in [0,1] \mid g(\beta) \le \tau\}$.

Note. Both g and g^{-1} are convex programs in one variable and can be efficiently computed (e.g., via line search). In certain divergence choices, closed-form solutions exist; for example, with the total-variation generator $f(r) = \frac{1}{2}|r-1|$, one obtains $g(\beta) = \max(0, \beta - r)$, which yields an explicit robustness adjustment.

Given Lemma 4, we can easily compute $\Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)})$ for the f-divergence as

$$\Delta_{\mathsf{d}}(r; R^{(1)}, \ldots, R^{(K)}) = C^{\text{rob}} - C.$$

(b) Lévy–Prokhorov–Based Robust Conformal Prediction

The Lévy–Prokhorov (LP) ambiguity set models distribution shifts by allowing both local (Wasserstein-type) and global (total-variation-type) perturbations and is defined as

$$\mathcal{U}_{\varepsilon,\rho}^{\mathsf{LP}}(\mathcal{R}_{\text{train}}) = \{\, \mathcal{Q} : \mathsf{LP}_\varepsilon(\mathcal{R}_{\text{train}}, \mathcal{Q}) \le \rho \,\}, \tag{15}$$

where the LP pseudo-metric is defined as

$$\mathsf{LP}_\varepsilon(\mathcal{R}_{\text{train}}, \mathcal{Q}) = \inf_{\gamma \in \Gamma(\mathcal{R}_{\text{train}}, \mathcal{Q})} \int \mathbf{1}\{\|z_1 - z_2\| > \varepsilon\}\, d\gamma(z_1, z_2).$$

Here, $\Gamma(\mathcal{R}_{\text{train}}, \mathcal{Q})$ denotes the set of all couplings with marginals $\mathcal{R}_{\text{train}}$ and $\mathcal{Q}$. As shown in Aolaritei et al. (2025), this ambiguity set can be decomposed as

$$\mathcal{U}_{\varepsilon,\rho}^{\mathsf{LP}}(\mathcal{R}_{\text{train}}) = \bigcup_{\widetilde{\mathcal{P}} : W_\infty(\mathcal{R}_{\text{train}}, \widetilde{\mathcal{P}}) \le \varepsilon} \{\, \mathcal{Q} : \text{TV}(\widetilde{\mathcal{P}}, \mathcal{Q}) \le \rho \,\},$$

highlighting that ε controls local geometric shifts (through W_∞) while ρ allows a ρ-fraction of global mass displacement. Similar to the f-divergence, the calibration scores from $\mathcal{R}_{\text{train}}$ can now be used to construct a robust quantile that guarantees coverage under all $\mathcal{R}_{\text{test}} \in \mathcal{U}_r^{\mathsf{LP}}$.

Lemma 5 (Corollary 4.2 from Aolaritei et al. (2025)) *Let $R^{(0)}, \dots, R^{(K)}$ be independent random variables with $R^{(0)} \sim \mathcal{R}_{\text{test}}$ and $R^{(1)}, \dots, R^{(K)} \sim \mathcal{R}_{\text{train}}$, satisfying $\mathcal{R}_{\text{test}} \in \mathcal{U}^{\mathsf{LP}}_{\varepsilon,\rho}(\mathcal{R}_{\text{train}})$. For a failure probability $\delta \in (0,1)$, define the set*

$$C^{\text{rob}} := \text{Quantile}_{1-\beta+\rho}\big(R^{(1)}, \dots, R^{(K)}\big) + \varepsilon$$

with $\beta = \delta + \frac{\delta-\rho-2}{K}$. Then, $\mathbb{P}\big(R^{(0)} \le C^{\text{rob}}\big) \ \ge \ 1-\delta$.

Looking at Lemma 5, we note that the robust quantile C^{rob} is obtained by inflating the standard quantile in two ways: (i) ρ shifts the quantile *level* to $1-\beta+\rho$, and (ii) ε introduces a uniform additive displacement reflecting the largest possible change in score induced by a local perturbation.

7.3. Generative Modeling and Conditional Diffusion Models

This section provides a concise overview of the diffusion modeling framework used in our approach. Specifically, we (a) briefly review the formulation of Denoising Diffusion Probabilistic Models (DDPMs), (b) introduce conditional diffusion models with context variables, and (c) describe the CARD architecture and its associated training objective. We follow standard DDPM constructions from Ho et al. (2020); Song et al. (2020) and their conditional extension developed in Han et al. (2022).

(a) Diffusion Models. Diffusion models define a generative process by progressively perturbing data samples $s_0 \sim p_{\text{data}}$ – drawn from an unknown underlying data distribution p_{data} – with Gaussian noise, and then learning to reverse this process to recover clean samples. The forward process constructs a sequence of latent variables $\{s_j\}_{j=0}^{T_{\text{diff}}}$ via a fixed variance schedule $\{\beta_j\}_{j=1}^{T_{\text{diff}}}$

$$q(s_j \mid s_{j-1}) = \mathcal{N}\Big(\sqrt{1-\beta_j}\, s_{j-1},\, \beta_j \mathbf{I}\Big)\,, \tag{16}$$

which admits a closed-form expression for any diffusion step

$$q(s_j \mid s_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_j}\, s_0,\, (1-\bar{\alpha}_j)\mathbf{I}\big) \tag{17}$$

where $\bar{\alpha}_j = \prod_{s=1}^{j}(1-\beta_s)$. The reverse (denoising) process approximates the true posterior $q(s_{j-1} \mid s_j)$ by a parameterized Gaussian:

$$p_\theta(s_{j-1} \mid s_j) = \mathcal{N}(\mu_\theta(s_j, j), \Sigma_j)\,, \tag{18}$$

where the mean is reparameterized using a neural network $\epsilon_\theta(s_j, j)$ that predicts the added noise:

$$\mu_\theta(s_j, j) = \frac{1}{\sqrt{\alpha_j}}\left(s_j - \frac{\beta_j}{\sqrt{1-\bar{\alpha}_j}}\,\epsilon_\theta(s_j, j)\right). \tag{19}$$

The reverse-process covariance is typically chosen as $\Sigma_j = \beta_j \mathbf{I}$. This construction defines a Markov chain that gradually transforms Gaussian noise into samples that are drawn from the learned data distribution p_θ and approximate $s_0 \sim p_{\text{data}}$. The process of generating new data then consists of sampling $x_{T_{\text{diff}}} \sim \mathcal{N}(0, \mathbf{I})$ and recursively applying (18).

(b) Conditional Diffusion Models (CARD design). To model structured dependencies or contextual information, diffusion models can be extended to represent a family of conditional distributions $p_\theta(s_0 \mid \mathbf{c})$, where $\mathbf{c}$ denotes conditioning covariates such as control inputs, environmental descriptors, or temporal indices. In the CARD framework (Han et al., 2022), the key distinction is that the *forward* diffusion becomes context-dependent via a pre-trained conditional mean encoder $f_\phi(\mathbf{c})$. Specifically, the forward process is modified to drift toward the context-dependent mean:

$$q(s_j \mid s_{j-1}, \mathbf{c}) = \mathcal{N}\Big(\sqrt{1-\beta_j}\, s_{j-1} + \big(1 - \sqrt{1-\beta_j}\big) f_\phi(\mathbf{c}),\ \beta_j \mathbf{I}\Big). \tag{20}$$

This modification ensures that the noisy variables remain centered around a context-informed trajectory, so that the reverse process learns deviations around a meaningful conditional mean rather than reconstructing it from scratch.

The reverse process then follows the DDPM parameterization but conditioned on $\mathbf{c}$:

$$p_\theta(s_{j-1} \mid s_j, \mathbf{c}) = \mathcal{N}\big(\mu_\theta(s_j, j, f_\phi(\mathbf{c})),\ \Sigma_j\big), \tag{21}$$

where the mean is expressed as

$$\mu_\theta(s_j, j, f_\phi(\mathbf{c})) = \frac{1}{\sqrt{\alpha_j}} \left(s_j - \frac{\beta_j}{\sqrt{1-\bar{\alpha}_j}}\, \epsilon_\theta(s_j, j, f_\phi(\mathbf{c})) \right). \tag{22}$$

The learnable parameters θ inhabit the noise-prediction network ϵ_θ, whereas $f_\phi(\mathbf{c})$ is provided by a pre-trained encoder. The reverse-process covariance Σ_j has the same definition as before, typically chosen as $\Sigma_j = \beta_j \mathbf{I}$.

(c) Training Objective. The noise-prediction network ϵ_θ is trained with a conditional DDPM loss adapted to the CARD forward process:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{s_0, \mathbf{c}, j, \varepsilon}\Big[\big\| \varepsilon - \epsilon_\theta\big(s_j, j, f_\phi(\mathbf{c})\big) \big\|_2^2 \Big],$$

where s_j is sampled from the CARD forward kernel

$$s_j = \sqrt{\bar{\alpha}_j}\, s_0 + (1 - \sqrt{\bar{\alpha}_j})\, f_\phi(\mathbf{c}) + \sqrt{1-\bar{\alpha}_j}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I).$$

The encoder f_ϕ is typically *pre-trained* (and then fixed during diffusion training) using a simple regression objective

$$\mathcal{L}_{\text{enc}}(\phi) = \mathbb{E}_{s_0, \mathbf{c}}\big[\| s_0 - f_\phi(\mathbf{c}) \|_2^2 \big],$$

so that $f_\phi(\mathbf{c}) \approx \mathbb{E}[s_0 \mid \mathbf{c}]$, using a (possibly disjoint) subset of the same data distribution employed to train the CARD model.

When the conditioning variable is omitted, the formulation reduces to the standard, unconditional diffusion model. Although the model is trained via noise regression, this objective is algebraically equivalent to denoising score matching for the DDPM/CARD forward process; the precise conversion used for the analytical bound is provided in Appendix 7.4.

7.4. Analytical Error Bound for CARD

For a given context $\mathbf{c}$, Li (2025) obtained a computable upper bound on the 2-Wasserstein distance

$$r = W_2\big(q(\cdot \mid \mathbf{c}),\, p_\theta(\cdot \mid \mathbf{c})\big),$$

where $q(\cdot \mid \mathbf{c})$ denotes the unknown conditional data distribution and $p_\theta(\cdot \mid \mathbf{c})$ is the conditional distribution learned by the CARD model.

Strategy and theoretical foundation. We rely on a result from Li (2025), which upper bounds the 2-Wasserstein distance between the two aforementioned distributions in terms of the score-matching error and certain regularity constants. To apply this result to the CARD model used in our framework, we proceed in three steps: (a) we relate the noise-matching training objective used in CARD to the score-matching objective that appears in the result from Li (2025); (b) we investigate the required regularity assumptions and explicitly compute the associated constants for the CARD dynamics; and (c) we specialize the resulting Wasserstein bound to our conditional setting and present it as a theorem tailored to our model.

(a) Equivalence of noise and score objectives. CARD trains a denoising network by regressing the additive noise ε (the ε-prediction or "noise" objective), whereas the analytical error bound in Li (2025) is formulated in terms of a score-matching objective. For DDPM-style forward processes, these two objectives are equivalent. We make this equivalence explicit below in order to connect the CARD training error to the quantity appearing in the Wasserstein error bound from Li (2025).

Forward marginal and notation. For both DDPM and CARD, the forward diffusion marginal at diffusion step j is Gaussian. To simplify notation, we introduce the shorthand[3]

$$\kappa_j(s_0, f_\phi(\mathbf{c})) = \sqrt{\bar{\alpha}_j}\, s_0 + (1 - \sqrt{\bar{\alpha}_j}) f_\phi(\mathbf{c})$$

which represents the context-dependent mean of the forward process. With this notation, the conditional density of s_j given $(s_0, \mathbf{c})$ is

$$q(s_j \mid s_0, \mathbf{c}) = \mathcal{N}\big(\kappa_j(s_0, f_\phi(\mathbf{c})),\, (1 - \bar{\alpha}_j)\mathbf{I}\big),$$

that is, the value of the Gaussian probability density with mean $\kappa_j(s_0, \mathbf{c})$ and covariance $(1 - \bar{\alpha}_j)\mathbf{I}$.

Score of the forward marginal. The *score* of a distribution is defined as the gradient of its log-density with respect to the variable of interest. For the Gaussian density above, this is

$$\log q(s_j \mid s_0, \mathbf{c}) = -\frac{1}{2(1 - \bar{\alpha}_j)} \|s_j - \kappa_j(s_0, f_\phi(\mathbf{c}))\|^2 + \text{const.}$$

Differentiating with respect to s_j yields the *score* as expressed below:

$$\nabla_{s_j} \log q(s_j \mid s_0, \mathbf{c}) = -\frac{s_j - \kappa_j(s_0, f_\phi(\mathbf{c}))}{1 - \bar{\alpha}_j}.$$

Sampling interpretation (reparameterization). The expression for $q(s_j \mid s_0, \mathbf{c})$ defined previously describes the distribution of s_j. Equivalently, a random sample s_j drawn from this Gaussian distribution can be written using the standard reparameterization of a normal random variable as

$$s_j = \kappa_j(s_0, f_\phi(\mathbf{c})) + \sqrt{1 - \bar{\alpha}_j}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \mathbf{I}).$$

3. Note that we introduce the expression κ for concise writing and has no other significance in related literature.

This representation does not introduce a new assumption; it is simply an explicit way of expressing samples from $\mathcal{N}(\kappa_j, (1-\bar{\alpha}_j)\mathbf{I})$. Substituting this expression for s_j into the score yields the key noise–score identity:

$$\nabla_{s_j} \log q(s_j \mid s_0, \mathbf{c}) = -\frac{1}{\sqrt{1-\bar{\alpha}_j}}\,\varepsilon.$$

From noise regression to score matching. As previously mentioned, CARD is trained by minimizing the noise regression error

$$\|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) - \varepsilon\|^2.$$

Using the noise-score identity above, the noise variable can be written in terms of the score as

$$\varepsilon = -\sqrt{1-\bar{\alpha}_j}\,\nabla_{s_j} \log q(s_j \mid s_0, \mathbf{c}).$$

Substituting this expression into the noise regression error yields

$$\|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) - \varepsilon\|^2 = \|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) + \sqrt{1-\bar{\alpha}_j}\,\nabla_{s_j} \log q(s_j \mid s_0, \mathbf{c})\|^2.$$

We now define the score estimator implicitly induced by the noise-prediction network as

$$s_\theta(s_j, j, f_\phi(\mathbf{c})) = -\frac{1}{\sqrt{1-\bar{\alpha}_j}}\,\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})).$$

If the model predicts the score $s_\theta(s_j, j, f_\phi(\mathbf{c}))$ or the noise $\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c}))$, the identity above implies the pointwise relation. With this definition, the error term becomes

$$\|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) - \varepsilon\|^2 = \| - \sqrt{1-\bar{\alpha}_j}\,\big(s_\theta(s_j, j, f_\phi(\mathbf{c})) - \nabla_{s_j} \log q(s_j \mid s_0, \mathbf{c})\big)\|^2.$$

Taking squared expectations on both sides gives

$$\mathbb{E}\big[\|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) - \varepsilon\|^2\big] = (1-\bar{\alpha}_j)\,\mathbb{E}\big[\|s_\theta(s_j, j, f_\phi(\mathbf{c})) - \nabla_{s_j} \log q(s_j \mid s_0, \mathbf{c})\|^2\big]\,.$$

Denoting the noise mean-squared error by $E(j) = \mathbb{E}[\|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) - \varepsilon\|^2]$ and the score mean-squared error by $H(j) = \mathbb{E}[\|s_\theta(s_j, j, \mathbf{c}) - \nabla \log q(s_j \mid s_0, f_\phi(\mathbf{c}))\|^2]$, we obtain the exact relation

$$H(j) = \frac{E(j)}{1-\bar{\alpha}_j}.$$

Thus, the noise-regression objective used to train CARD is exactly equivalent, up to a known scaling factor, to the score-matching error required by the Wasserstein error bound from Li (2025). Therefore, inserting H_j into (Li, 2025, Theorem 1) results directly in Theorem 6 (stated below) by translating the score-regression quantity into a noise-regression empirical error.

(b) Assumptions and regularity conditions. The analytical bound of Li (2025) requires mild regularity conditions on the diffusion dynamics and the associated score functions. In particular, the bound depends on two step-dependent Lipschitz constants: $L_1(j)$ (associated with the forward diffusion drift) and $L_2(j)$ (associated with the score function) where $j \in \{1, \ldots, T_{\text{diff}}\}$ denotes the diffusion step for a total of T_{diff} diffusion steps. The dependence on the diffusion step j arises naturally because both the drift and the score vary across noise levels in the diffusion process.

Specifically, the assumptions are: (i) a one-sided Lipschitz condition on the forward drift $b_j(s) = -\frac{1}{2}\beta_j(s - f_\phi(\mathbf{c}))$ with constant $L_1(j)$; (ii) a one-sided Lipschitz condition on both the true and learned score functions with constant $L_2(j)$; (iii) a bounded diffusion variance schedule $\{\beta_j\}$; and (iv) consistency of the pre-trained conditional mean encoder, namely that for any $\delta > 0$ there exists a_0 such that for all training epochs $a > a_0$, $\mathbb{P}(\|f_\phi(\mathbf{c}) - \mathbb{E}[s_0 \mid \mathbf{c}]\|_2 < \delta) \to 1$. These conditions match the regularity assumptions required in Li (2025).

Analytical form of the drift Lipschitz constant $L_1(j)$. The analytical bound of Li (2025) assumes that the forward drift satisfies a Lipschitz-type regularity condition. In the CARD formulation, the forward diffusion drift at diffusion step j is given by $b_j(s) = -\frac{1}{2}\beta_j\big(s - f_\phi(\mathbf{c})\big)$. For any two states $s_a, s_b \in \mathbb{R}^d$, we have $\|b_j(s_a) - b_j(s_b)\|^2 = \big\|-\frac{1}{2}\beta_j(s_a - s_b)\big\|^2 = \frac{\beta_j^2}{4}\|s_a - s_b\|^2$. Thus, the drift satisfies the squared-norm Lipschitz inequality $\|b_j(s_a) - b_j(s_b)\|^2 \le L_1(j)\,\|s_a - s_b\|^2$, with the explicit analytical constant $L_1(j) = \frac{\beta_j^2}{4}$ which we use in our experiments. This verifies the required regularity condition for the forward diffusion drift used in the Wasserstein bound.

Estimation of the score Lipschitz constant $L_2(j)$. Unlike the drift, the score function is represented by a neural network s_θ and is therefore not available in closed form. The constant $L_2(j)$ quantifies the one-sided Lipschitz continuity of the learned score function $s_\theta(s_j, j, f_\phi(\mathbf{c}))$ and is defined through the inequality

$$\big(s_\theta(s_a, j, f_\phi(\mathbf{c})) - s_\theta(s_b, j, f_\phi(\mathbf{c}))\big)^\top (s_a - s_b) \le L_2(j)\,\|s_a - s_b\|^2,$$

for all noisy states s_a, s_b at diffusion step j under context $\mathbf{c}$. In our experiments, we estimate $L_2(j)$ empirically. Therefore, we sample pairs of noisy states (s_a, s_b) are sampled from the forward marginal $q(s_j \mid s_0, \mathbf{c})$, evaluate the corresponding score network outputs, and compute the ratio

$$\frac{\big(s_\theta(s_a, j, f_\phi(\mathbf{c})) - s_\theta(s_b, j, f_\phi(\mathbf{c}))\big)^\top (s_a - s_b)}{\|s_a - s_b\|^2}.$$

Subsequently, we choose the maximum value of this quantity over all the random pairs to obtain an estimate of $L_2(j)$.

(c) The Wasserstein error bound. With the previous definitions and explanations, we are now ready to state (Li, 2025, Theorem 1), which we have used in the main part of the paper.

Theorem 6 (Analytical 2-Wasserstein Bound, Li (2025)) *Let $q(\cdot \mid \mathbf{c})$ denote the true conditional data distribution and $p_\theta(\cdot \mid \mathbf{c})$ the conditional distribution induced by a diffusion-based generative model trained via denoising or score matching. Suppose the drift and score functions satisfy one-sided Lipschitz and smoothness conditions with constants $L_1(j)$ and $L_2(j)$. Then, the 2-Wasserstein distance between the true and the learned conditional distribution satisfies*

$$W_2\big(q(\cdot \mid \mathbf{c}),\, p_\theta(\cdot \mid \mathbf{c})\big) \;\le\; \underbrace{\sum_{j=1}^{T_{\text{diff}}} \beta_j\, M(j)\, \sqrt{H(j)}}_{\textit{score approximation error}} \;+\; \underbrace{\left(\sum_{j=1}^{T_{\text{diff}}} \beta_j^2 \exp\Big(\sum_{s \le j}\big(L_1(s) + L_2(s)\beta_s\big)\Big)\right)^{1/2}}_{\textit{residual discretization term}}, \tag{23}$$

where $H(j) = E(j)/(1 - \bar{\alpha}_j)$ and $E(j) = \mathbb{E}\big[\|\varepsilon_\theta(s_j, j, f_\phi(\mathbf{c})) - \varepsilon\|^2\big]$ is the per-step noise MSE, $M(j) = \exp\Big(\sum_{s \le j}(L_1(s) + L_2(s)\beta_s)\Big)$, and β_t are the diffusion noise coefficients.

We note that under the regularity assumptions of Li (2025), the residual discretization term vanishes as $T_{\text{diff}} \to \infty$ or for sufficiently small β_t, reducing the bound to the dominant first term which we stated in the main part of the paper. Finally, to map Theorem 6 to our setting in the main part of the paper, we can simply view $\bar{\mathcal{R}}_{\text{test}}(\zeta)$ as $q(\cdot \mid \mathbf{c})$.

7.5. Analytical Robustification of the Conformal Threshold

This section explains how the Wasserstein error bound from Theorem 6 for the conditional diffusion model can be used to obtain $C^{\text{rob}}_{\tau|t,i}$ that makes Lemma 2 valid. The argument proceeds in three steps: (a) we recall how to obtain Wasserstein bounds of Lipschitz continuous functions, (b) we relate Wasserstein bounds from the model p_θ to Wasserstein bounds of the nonconformity scores, and (c) we use the resulting bound to construct the constant $C^{\text{rob}}_{\tau|t,i}$ that makes Lemma 2 valid.

(a) Wasserstein bounds of Lipschitz continuous functions. Let $\mathcal{D}$ and $\mathcal{D}_0$ denote two probability distributions, where $\mathcal{D}$ could represent the true underlying distribution while $\mathcal{D}_0$ could represent the learned distribution, approximating $\mathcal{D}$. Let f be an L-Lipschitz continuous function with respect to the underlying metrics that, if applied to samples from $\mathcal{D}$ and $\mathcal{D}_0$, transforms the distributions $\mathcal{D}$ and $\mathcal{D}_0$ into the pushforward distributions $\mathcal{R}$ and $\mathcal{R}_0$, respectively. Existing bounds for L-Lipschitz continuous functions (see, e.g., Villani (2008)) then guarantee that

$$W_2(\mathcal{R}, \mathcal{R}_0) \;\leq\; L\, W_2(\mathcal{D}, \mathcal{D}_0).$$

(b) Propagation of Wasserstein bound to the nonconformity score distribution. Recall now that the original nonconformity score in Section 4.1 was defined as

$$R^{(k)} = \max_{(t,\tau,i)\in\mathcal{S}} \frac{\|\hat{Y}^{(k)}_{\tau|t,i} - Y^{(k)}_{\tau,i}\|}{\sigma_{\tau|t,i}}. \tag{24}$$

In Section 4.2 we learned a conditional diffusion model $p_\theta(\bar{R} \mid \mathbf{c})$ that predicts $\|\hat{Y}^{(k)}_{\tau|t,i} - Y^{(k)}_{\tau,i}\|$ for trajectories $Y^{(k)}$. This way, we approximate the nonconformity score in (24) as

$$R^{(k)} = \max_{(t,\tau,i)\in\mathcal{S}} \frac{\bar{R}^{(k)}_{\tau|t,i}}{\sigma_{\tau|t,i}} \tag{25}$$

where $\bar{R}^{(k)}_{\tau|t,i}$ is generated by $p_\theta(\bar{R} \mid \mathbf{c})$.

Suppose next that the distribution of the learned conditional generative model $p_\theta(\bar{R} \mid \mathbf{c})$ corresponding to $\bar{R}^{(k)}_{\tau|t,i}$ satisfies a Wasserstein bound of ε_W with the distribution $\bar{\mathcal{R}}_{\text{test}}(\zeta)$ corresponding to $\|\hat{Y}^{(k)}_{\tau|t,i} - Y^{(k)}_{\tau,i}\|$, e.g., obtained using Theorem 6. This means that

$$W_2\left(\bar{\mathcal{R}}_{\text{test}}(\zeta), p_\theta(\bar{R} \mid \mathbf{c})\right) \leq \varepsilon_W.$$

Next note that the function $f(s) = s/\sigma_{\tau|t,i}$ is $1/\sigma_{\tau|t,i}$-Lipschitz continuous. Similarly, the function $f(s) = \max_{(t,\tau,i)\in\mathcal{S}} s/\sigma_{\tau|t,i}$ is $1/\sigma_{\min}$-Lipschitz continuous where $\sigma_{\min} := \min_{(t,\tau,i)\in\mathcal{S}} \sigma_{\tau|t,i}$. Finally, let $\bar{\mathcal{R}}_{\text{test}}(\zeta)$ and $p_\theta(\bar{R} \mid \mathbf{c})$ correspond to $\mathcal{D}$ and $\mathcal{D}_0$, respectively, while $\mathcal{R}$ and $\mathcal{R}_0$ follow

equations (24) and (25) so that they are induced by $\mathcal{D}$ and $\mathcal{D}_0$, respectively. Then, we can use the previous Wasserstein bounds for Lipschitz continuous functions and conclude that

$$W_2(\mathcal{R}, \mathcal{R}_0) \leq \varepsilon_W / \sigma_{\min}.$$

By this result it follows that any Wasserstein error of ε_W in the data distribution induces at most a proportional Wasserstein error of $\varepsilon_W/\sigma_{\min}$ in the resulting nonconformity score distribution. This result provides a principled mechanism to quantify how uncertainty in the data-generating process propagates through the conformal mapping.

(c) Robust conformal threshold. Let us now interpret $\mathcal{R}_{\text{test}}$ and $\mathcal{R}_{\text{train}}$ from Lemma 2 as $\mathcal{R}$ and $\mathcal{R}_0$, respectively. In this case, we know that r in Lemma 2 is $\varepsilon_W/\sigma_{\min}$ so that the choice of

$$C^{\text{rob}}_{\tau|t,i} := \sigma_{\tau|t,i}\Big(C + \Delta_{\mathsf{d}}\Big(\tfrac{\varepsilon_W}{\sigma_{\min}};\, R^{(1)}, \ldots, R^{(K)}\Big)\Big) \tag{26}$$

makes Lemma 2 valid.

7.6. Time Complexity Analysis in Case Study

At each MPC timestep, our approach adds computation due to synthetic sample generation for the diffusion model, beyond solving the MPC problem. The baseline MPC requires $\sim 42\,\text{ms}$ per step, while incorporating sampling increases this cost (Fig. 4) but remains well below $1\,\text{s}$. For $K = 500$–1000 samples (sufficient for $\delta = 0.1$), the average computation time stays under $0.5\,\text{s}$, demonstrating practical real-time feasibility with calibrated uncertainty guarantees.

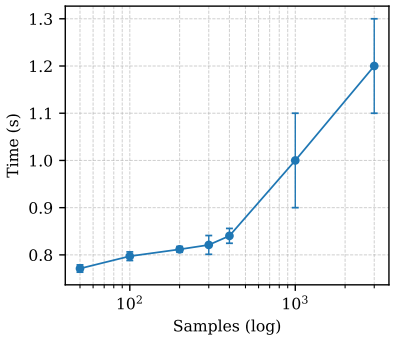

Figure 4: Average computation time per MPC step versus the number of synthetic samples K (log scale), including both MPC optimization and synthetic sample generation.