



# Signal temporal logic control synthesis among uncontrollable dynamic agents with conformal prediction<sup>☆</sup>

Xinyi Yu<sup>a,\*</sup>, Yiqi Zhao<sup>a</sup>, Xiang Yin<sup>b</sup>, Lars Lindemann<sup>c</sup>

<sup>a</sup> Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

<sup>b</sup> The School of Automation and Sensing, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>c</sup> The Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland

## ARTICLE INFO

### Article history:

Received 8 October 2024

Received in revised form 14 May 2025

Accepted 30 July 2025

### Keywords:

Signal temporal logic control

Control among dynamic stochastic agents

Conformal prediction

## ABSTRACT

The control of dynamical systems under temporal logic specifications among uncontrollable dynamic agents is challenging due to the agents' a-priori unknown behavior. Existing works have considered the problem where either all agents are controllable, the agent models are deterministic and known, or no safety guarantees are provided. We propose a predictive control synthesis framework that guarantees, with high probability, the satisfaction of signal temporal logic (STL) tasks that are defined over a controllable system in the presence of uncontrollable stochastic agents. We use trajectory predictors and conformal prediction to construct probabilistic prediction regions for each uncontrollable agent that are valid over multiple future time steps. Specifically, we construct a normalized prediction region over all agents and time steps to reduce conservatism and increase data efficiency. We then formulate a worst-case bilevel mixed integer program (MIP) that accounts for all agent realizations within the prediction region to obtain an open-loop controller that provably guarantee task satisfaction with high probability. To efficiently solve this bilevel MIP, we propose an equivalent MIP program based on KKT conditions of the original bilevel formulation. Building upon this, we design a closed-loop controller, where both recursive feasibility and task satisfaction can be guaranteed with high probability. We illustrate our control synthesis framework on two case studies.

© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

## 1. Introduction

Consider the following scenarios in which autonomous dynamical systems need to operate safely among uncontrollable dynamical agents: mobile service robots operating in pedestrian-rich environments, self-driving cars navigating through dense urban traffic, and leader-follower systems such as truck platoons reducing air drag to improve fuel economy. All these scenarios have in common that the autonomous system needs to satisfy complex safety-critical specifications in the presence of uncontrollable agents. Achieving these specifications is challenged by the fact that communication is prohibited, limited and unstable, or only one-directional, while the agents' intentions may not always be known. In this setting, the autonomous system needs

to make predictions about the behavior of the dynamic agents and select safe actions accordingly. In this work, we propose a predictive control synthesis framework with probabilistic safety for complex temporal logic specifications.

Control of multi-agent systems under temporal logic specifications has attracted much attention, see e.g., Büyükoçak, Aksaray, and Yazicioglu (2021), Guo and Dimarogonas (2015), Kantaros, Guo, and Zavlanos (2019), Kantaros and Zavlanos (2020), Kloetzer and Belta (2009), Lindemann and Dimarogonas (2019), Srinivasan, Coogan, and Egerstedt (2018), Sun, Chen, Mitra, and Fan (2022). However, the agent dynamics here are usually known and agents are assumed to be collaborative. In this paper, we are interested in control synthesis among uncontrollable dynamic stochastic agents under temporal logic tasks, and specifically in settings where communication is limited and no information about agent intentions and models is available. This problem has been studied in the literature, e.g., using sampling-based approaches (Hoxha & Fainekos, 2016; Kalluraya, Pappas, & Kantaros, 2023), game theory (Li, Shahrivar, & Liu, 2021), supervisory control theory (Kloetzer & Mahulea, 2012), or optimization-based methods (Ayala, Andersson, & Belta, 2011; Purohit & Saha, 2021; Ulusoy & Belta, 2014). These works, however, only consider qualitative temporal logic specifications such as linear temporal logic, while we

<sup>☆</sup> This work was supported in part by Annenberg Fellowship, Viterbi School of Engineering Fellowship, National Science Foundation through the grant IIS-SLES-2417075, and National Natural Science Foundation of China 62173226. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Yongcan Cao under the direction of Editor Christos G. Cassandras.

\* Corresponding author.

E-mail addresses: [xinyi.yu12@usc.edu](mailto:xinyi.yu12@usc.edu) (X. Yu), [yiqizhao@usc.edu](mailto:yiqizhao@usc.edu) (Y. Zhao), [yinxiang@sjtu.edu.cn](mailto:yinxiang@sjtu.edu.cn) (X. Yin), [llindemann@ethz.ch](mailto:llindemann@ethz.ch) (L. Lindemann).

are here interested in quantitative temporal logics that allow to measure the quality of task satisfaction. More importantly, these works make simplifying assumptions on the models of the agents such as being linear or having Gaussian noise distributions. Recent works in [Dixit, Lindemann, Wei, Cleaveland, Pappas, and Burdick \(2023\)](#), [Lindemann, Cleaveland, Shim, and Pappas \(2023a\)](#), [Muthali et al. \(2023\)](#), [Tonkens, Sun, Yu, and Herbert \(2023\)](#) make no such assumptions and instead use learning-enabled trajectory predictors and conformal prediction to obtain probabilistic prediction regions that can be used for downstream control. However, these works do not consider complex specifications expressed in temporal logics. Note that extending the control frameworks aforementioned to the temporal logic case is not straightforward, since temporal logic specifications are defined over agent trajectories while the aforementioned works only consider point-wise state constraints.

**Contribution.** We consider the signal temporal logic (STL) control synthesis problem in the presence of uncontrollable dynamic agents without making any assumptions on the agents' dynamics and distribution. Specifically, we make the following contributions:

- We use trajectory predictors and conformal prediction to construct probabilistic prediction regions that contain the trajectories of multiple uncontrollable agents.
- We propose a new mixed integer program (MIP) based on a worst case formulation of the STL constraints over all uncontrollable agent trajectories within these prediction regions. We present an equivalent MIP program based on KKT conditions of the original MIP to efficiently compute the open-loop controller that provably guarantee task satisfaction with high probability.
- To obtain a closed-loop controller, we introduce a recursively feasible model predictive control (MPC) framework, where recursive feasibility through the task horizon can be guaranteed with high probability. We guarantee task satisfaction with high probability.
- We illustrate the efficacy of our algorithms on temperature control and robot motion planning problems.

All proofs of our technical results can be found in [Yu, Zhao, Yin, and Lindemann \(2023\)](#) in order to save space.

### 1.1. Related work

Quantitative temporal logics such as signal temporal logic (STL) ([Maler & Nickovic, 2004](#)) have increasingly been used in control. The reason for their success in control is their ability to reason over the robustness of a control system with respect to the specification via the STL quantitative semantics ([Donzé & Maler, 2010a](#); [Fainekos & Pappas, 2009](#)). The quantitative semantics of STL provide a natural control objective that can be maximized, e.g., via gradient-based optimization ([Gilpin, Kurtz, & Lin, 2020](#); [Mehdipour, Vasile, & Belta, 2019](#); [Pant, A., & Mangharam, 2017](#); [Pant, Abbas, Quaye, & Mangharam, 2018](#)), mixed integer programming ([Kurtz & Lin, 2022](#); [Raman, Donzé, Maa-soumy, Murray, Sangiovanni-Vincentelli, & Seshia, 2014](#); [Sadraddini & Belta, 2018](#)), or control barrier functions ([Charitidou & Dimarogonas, 2022](#); [Lindemann & Dimarogonas, 2018](#); [Xiao, Belta, & Cassandras, 2021](#)).

While STL control synthesis problems have been well studied for deterministic systems, the problem becomes more challenging for stochastic systems. There are various approaches that ensure probabilistic task satisfaction, e.g., via mixed integer programming ([Farahani, Majumdar, Prabhu, & Soudjani, 2017](#); [Jha,](#)

[Raman, Sadigh, & Seshia, 2018](#); [Sadigh & Kapoor, 2016](#)), sampling-based approaches ([Ilyes, Ho, & Lahijanian, 2023](#); [Scher, Sadraddini, Yadin, & Kress-Gazit, 2023](#); [Vasile, Raman, & Karaman, 2017](#)), and stochastic control barrier functions ([Jagtap, Soudjani, & Zamani, 2020](#)). Another recent line of work considers the minimization of mathematical risk of STL specifications ([Akella, Dixit, Ahmadi, Burdick, & Ames, 2024](#); [Hashemi et al., 2023](#); [Safaoui, Lindemann, Dimarogonas, Shames, & Summers, 2020](#)). In addition, there are also some related approaches for reactive STL control synthesis, see e.g., [Gundana and Kress-Gazit \(2021, 2022\)](#), [Lindemann, Pappas, and Dimarogonas \(2021\)](#), [Raman, Donzé, Sadigh, Murray, and Seshia \(2015\)](#).

Conceptually closest to our work are ([Dixit et al., 2023](#); [Lindemann et al., 2023a](#); [Muthali et al., 2023](#)) where the authors apply conformal prediction to obtain probabilistic prediction regions of uncontrollable dynamic stochastic agents. Conformal prediction is a lightweight statistical tool for uncertainty quantification, originally introduced in [Shafer and Vovk \(2008\)](#), [Vovk, Gammerman, and Shafer \(2005\)](#), that has recently attracted attention within the machine learning community, see e.g., [Angelopoulos and Bates \(2021\)](#), [Fontana, Zeni, and Vantini \(2023\)](#) for up-to-date tutorials. Compared to [Dixit et al. \(2023\)](#), [Lindemann et al. \(2023a\)](#), [Muthali et al. \(2023\)](#), we consider complex signal temporal logic specifications, that are defined over agent trajectories as opposed to considering simple state constraints. Previous works hence do not apply to our settings as we have to account for state couplings at different times. Furthermore, complex temporal logic tasks require the data-efficient computation of non-conservative probabilistic prediction regions to promote long horizon control with multiple agents. We therefore propose an efficient method for computing prediction regions, and formulate a robust control problem which is fundamentally different and less conservative from Lipschitz-based formulations proposed in the aforementioned works.

## 2. Preliminaries & problem definition

We consider a autonomous systems, e.g., a robotic system or an autonomous car, that is described by the discrete-time dynamics

$$x_{k+1} = f(x_k, u_k), \quad x_0 \in \mathcal{X} \quad (1)$$

where the function  $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$  is defined over the state and input domains  $\mathcal{X}$  and  $\mathcal{U}$ , respectively. Here,  $x_0$  is the known initial state and  $x_k \in \mathcal{X} \subset \mathbb{R}^{n_x}$  and  $u_k \in \mathcal{U} \subset \mathbb{R}^m$  are the  $n_x$ -dimensional state and the  $m$ -dimensional control input at time  $k$ , respectively. To synthesize control sequences later, let us also define  $\mathbf{u}_{k:T-1} := u_k \dots u_{T-1}$  and  $\mathbf{x}_{k+1:T} := x_{k+1} \dots x_T$ .

The system in Eq. (1) operates among  $N$  uncontrollable and stochastic dynamic agents, e.g., autonomous or remote-controlled robots or humans. We model the state of agent  $i$  at time  $k$  as a random variable  $Y_{k,i} \in \mathcal{Y}_i \subset \mathbb{R}^{n_{y,i}}$  with state domain  $\mathcal{Y}_i$ , and we define the random state of all agents as  $Y_k := [Y_{k,1}, \dots, Y_{k,N}] \in \mathcal{Y} \subset \mathbb{R}^{n_y}$  with  $n_y := n_{y,1} + \dots + n_{y,N}$  and  $\mathcal{Y} := \mathcal{Y}_1 \times \dots \times \mathcal{Y}_N$ . We do not assume to have any a-priori knowledge of the agent dynamics and intentions, and since the agents are uncontrollable, their finite-length trajectories are a-priori unknown. But we assume that trajectories follow an unknown distribution  $\mathcal{D}$ , i.e.,  $Y := (Y_0, Y_1, \dots) \sim \mathcal{D}$ . We further assume to have access to  $\bar{K}$  realizations from  $\mathcal{D}$ .

**Assumption 1.** We have access to  $\bar{K}$  independent trajectories  $Y^{(j)} := (Y_1^{(j)}, Y_2^{(j)}, \dots)$  from the distribution  $\mathcal{D}$ . We split the trajectories into calibration and training sets  $D_{cal} := \{Y^{(1)}, \dots, Y^{(K)}\}$  and  $D_{train} := \{Y^{(K+1)}, \dots, Y^{(\bar{K})}\}$ , respectively.

Modeling dynamic agents by a distribution  $\mathcal{D}$  includes general classes of systems such as discrete-time stochastic systems and Markov decision processes. We make the restriction that  $\mathcal{D}$  does not depend on the behavior of the autonomous system in (1), i.e.,  $\mathcal{D}$  does not depend on  $x$ . Similar modeling assumptions have been made in Lindemann et al. (2023a) and are justified in many applications, e.g., when conservative control actions are taken in autonomous driving that do only alter the behavior of pedestrians minimally. By using robust or adaptive versions of conformal prediction, see e.g., Cauchois, Gupta, Ali, and Duchi (2024), Gibbs and Candes (2021), we can relax this assumption. Lastly, Assumption 1 is not restrictive in most applications, e.g., past trajectory data of dynamic agents is available in robotic or transportation systems.

In this paper, we consider the control of the system in (1) under temporal logic specifications that are jointly defined over the system and the uncontrollable agents. Therefore, we define their joint state at time  $k$  as  $S_k := [x_k, Y_k] \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^n$  with  $n := n_x + n_y$  and the joint state sequence as  $\mathbf{S} := S_0, S_1, \dots$

### 2.1. Signal temporal logic control synthesis among uncontrollable agents

In this paper, we use signal temporal logic to describe the tasks defined over the system in (1) and uncontrollable agents. Signal temporal logic was originally introduced in Maler and Nickovic (2004), and we here only give a brief introduction. The syntax of discrete-time finite-length STL formulae is

$$\phi ::= \top \mid \pi^\mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathbf{U}_{[a,b]} \phi_2, \quad (2)$$

where  $\top$  is the true symbol ( $\perp$  denotes the false symbol) and  $\pi^\mu : \mathbb{R}^n \rightarrow \{\top, \perp\}$  is a predicate whose truth value is determined by the sign of an underlying predicate function  $\mu : \mathbb{R}^n \rightarrow \mathbb{R}$ , i.e., for the state  $s \in \mathbb{R}^n$  we have that  $\pi^\mu(s) := \top$  if and only if  $\mu(s) \geq 0$ . The symbols  $\neg$  and  $\wedge$  denote the standard Boolean operators “negation” and “conjunction”, respectively, and  $\mathbf{U}_{[a,b]}$  is the temporal operator “until” with  $a \leq b$ ,  $a, b \in \mathbb{N}$  with  $\mathbb{N}$  being the natural numbers. As we consider a discrete-time setting, we use  $[a, b]$  to denote the discrete-time interval  $[a, b] \cap \mathbb{N}$ . These operators can be used to define “disjunction” by  $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$ , “implication” by  $\phi_1 \rightarrow \phi_2 := \neg\phi_1 \vee \phi_2$ , “eventually” by  $\mathbf{F}_{[a,b]}\phi := \top \mathbf{U}_{[a,b]}\phi$ , and “always” by  $\mathbf{G}_{[a,b]}\phi := \neg\mathbf{F}_{[a,b]}\neg\phi$ .

**STL Semantics and Quantitative Semantics.** STL formulae are evaluated over trajectories  $\mathbf{s} := s_0 s_1 \dots$  where  $\mathbf{s}$  could be a realization of  $\mathbf{S}$ . The notation  $(\mathbf{s}, k) \models \phi$  denotes that  $\mathbf{s}$  satisfies the STL formula  $\phi$  at time  $k$ . In addition to these *qualitative* Boolean semantics, we can define *quantitative* semantics  $\rho^\phi(\mathbf{s}, k) \in \mathbb{R}$ , referred to as robust semantics, that describe how well  $\phi$  is satisfied (Donzé & Maler, 2010b). We note that  $\rho^\phi(\mathbf{s}, k) > 0$  implies  $(\mathbf{s}, k) \models \phi$ . The formal definition of these semantics can be found in Yu et al. (2023).

**Problem Formulation.** We consider bounded STL formulae  $\phi$  where time intervals  $[a, b]$  in  $\phi$  are bounded. The satisfaction of a bounded STL formula  $\phi$  can be computed over bounded trajectories. Specifically, we require trajectories with a minimum length of  $T_\phi$  where  $T_\phi$  is the formula horizon formally defined in Dokhanchi, Hoxha, and Fainekos (2014), Sadraddini and Belta (2015). For example, for  $\phi := \mathbf{F}_{[3,8]}\mathbf{G}_{[1,2]}\pi^\mu$ , we have  $T_\phi = 10$ . We additionally assume that  $\phi$  is in positive normal form, i.e., the formula  $\phi$  contains no negations, and is build from convex and differentiable predicate functions  $\mu$ . The former assumption is without loss of generality as every STL formula can be rewritten into an equivalent STL formula that is in positive normal form (Sadraddini & Belta, 2015, 2018). The latter assumption is made for computational reasons and can be relaxed, since every STL formula can be transformed into positive normal form (see Sadraddini and Belta (2015) for more details). We summarize these assumptions next.

**Assumption 2.** We consider bounded STL formulae  $\phi$  in positive normal form. We further assume that all predicate functions  $\mu$  in  $\phi$  are convex and differentiable with respect to  $Y_\tau$ .

We impose specifications  $\phi$  that satisfy Assumption 2 over the random trajectory  $\mathbf{S}$ , i.e., over the system trajectory  $x_k$  and the random trajectories of uncontrollable agents  $Y_k$ . We consider the following problem.

**Problem 1.** Given the system in (1), the uncontrollable agents  $Y := (Y_0, Y_1, \dots) \sim \mathcal{D}$ , a set of trajectories  $D_{cal}$  and  $D_{train}$  that satisfy Algorithm 1, an STL formula  $\phi$  from (2) that satisfies Assumption 2, and a failure probability  $\delta \in (0, 1)$ , compute the control inputs  $u_k$  for all  $k \in \{0, \dots, T_\phi - 1\}$  s.t. the STL formula is satisfied with a probability of at least  $1 - \delta$ , i.e.,  $\text{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta$ .

The challenge for solving Problem 1 is the dependency of  $\phi$  on the agents in  $Y$ , i.e., the joint trajectory  $\mathbf{S}$  is only partially controllable. Our approach is to predict uncontrollable agents, construct prediction regions that capture uncertainty of these predictions, and synthesize the controller using this information. While our work is inspired by Dixit et al. (2023), Lindemann et al. (2023a), these works cannot be simply extended to solve our problem, since they consider state constraints of the form  $c(x_k, y_k) \geq 0$  that are enforced at each time step separately. We instead consider STL specifications defined over the trajectory  $\mathbf{S}$ , where the task description cannot be separated into different time steps. This temporal coupling makes the problem of designing closed-loop controllers different from existing works, since during the online execution we cannot discard constraints from previous time steps. Furthermore, applying a Lipschitz reformulation as in Dixit et al. (2023), Lindemann et al. (2023a) will bring conservative results. These works leverage the Lipschitz constant of the function  $c$  to capture the worst case over the prediction region. In the STL case, taking the Lipschitz constant of the robustness function  $\rho^\phi$  may be conservative and not practical.

Our solution consists of an offline and an online phase. In the offline phase, we train trajectory predictors using historical data and compute prediction regions via conformal prediction (Section 3). These regions capture the uncertainty of uncontrollable agents' trajectories with a probability of at least  $1 - \delta$ . In the online phase, for open-loop control, we solve a mixed integer program (MIP) at the initial time step using the open-loop prediction regions, ensuring probabilistic satisfaction of the STL specification (Section 4). For closed-loop control, we iteratively update the predictions and prediction regions at each time step, solving the MIP at each time step and guaranteeing recursive feasibility and task satisfaction with high probability (Section 5).

### 2.2. Trajectory predictors and conformal prediction

**Trajectory Predictors.** From the training set  $D_{train}$ , we first train a trajectory predictor that estimates future states  $(Y_{k+1}, \dots, Y_{T_\phi})$  from past observations  $(Y_0, \dots, Y_k)$ . We denote the predictions made at time  $k$  as  $(\hat{Y}_{k+1|k}, \dots, \hat{Y}_{T_\phi|k})$ . For instance, we can use recurrent neural networks (RNNs) (Salehinejad, Sankar, Barfett, Colak, & Valaee, 2017), long short term memory (LSTM) networks (Yu, Si, Hu, & Zhang, 2019), or transformers (Han, Xiao, Wu, Guo, Xu, & Wang, 2021). In this paper, we do not make assumption on the trajectory predictor.

**Conformal Prediction.** The predictions  $\hat{Y}_{\tau|k}$  of  $Y_\tau$  for  $\tau > k$ , which may be obtained from learning-enabled components, may not always be accurate. We will use conformal prediction to obtain prediction regions for  $Y_\tau$  that are valid with high probability. We leverage conformal prediction which is a statistical tool for



uncertainty quantification with minimal assumptions (Shafer & Vovk, 2008; Vovk et al., 2005).

Let  $R^{(0)}, \dots, R^{(K)} \in \mathbb{R}$  be  $K + 1$  i.i.d. random variables which are referred to as the *nonconformity score*. For instance, we may define the prediction error  $R^{(j)} := \|Y_{\tau}^{(j)} - \hat{Y}_{\tau|k}^{(j)}\|$  at time  $\tau > k$  for all calibration trajectories  $Y^{(j)} \in D_{cal}$ . Naturally, a small value of  $R^{(j)}$  in this case implies accurate predictions. We later present nonconformity scores to obtain prediction regions for uncontrollable agents over multiple time steps. Given a failure probability  $\delta \in (0, 1)$ , our goal is to compute  $C$  from  $R^{(1)}, \dots, R^{(K)}$  such that the probability of  $R^{(0)}$  being bounded by  $C$  is not smaller than  $1 - \delta$ , i.e., such that

$$\text{Prob}(R^{(0)} \leq C) \geq 1 - \delta.$$

By the results in Lemma 1 of Tibshirani, Foygel Barber, Candes, and Ramdas (2019), we can compute

$$C := \text{Quantile}_{1-\delta}(R^{(1)}, \dots, R^{(K)}, \infty)$$

as the  $(1 - \delta)$ th quantile over the empirical distribution of  $R^{(1)}, \dots, R^{(K)}$  and  $\infty$ .<sup>1</sup> By assuming that  $R^{(1)}, \dots, R^{(K)}$  are sorted in non-decreasing order and by adding  $R^{(K+1)} := \infty$ , we have that  $C = R^{(p)}$  where  $p := \lceil (K + 1)(1 - \delta) \rceil$ , i.e.,  $C$  is the  $p$ th smallest nonconformity score. To obtain meaningful prediction regions, we note that  $K \geq \lceil (K + 1)(1 - \delta) \rceil$  has to hold. Otherwise, we get  $C = \infty$ . Lastly, note that the guarantees  $\text{Prob}(R^{(0)} \leq C) \geq 1 - \delta$  are marginal over the randomness in  $R^{(0)}, R^{(1)}, \dots, R^{(K)}$  ( $C$  depending on  $R^{(1)}, \dots, R^{(K)}$ ) and not conditional on  $R^{(1)}, \dots, R^{(K)}$ .

### 3. Open-loop prediction regions for multiple uncontrollable agents

To solve Problem 1, we use trajectory predictions and compute probabilistic prediction regions for each uncontrollable agent that are valid over multiple time steps. Recall that  $Y_{\tau,i}$  denotes the random state of agent  $i$  at time  $\tau$ , and let  $\hat{Y}_{\tau|0,i}$  denote the prediction of  $Y_{\tau,i}$  made at time  $k = 0$ . We are interested in constructing prediction regions for all predictions  $\tau \in \{1, \dots, T_{\phi}\}$  made at time  $k := 0$  and for all agents  $i \in \{1, \dots, N\}$ , i.e., s.t.

$$\text{Prob}(\|Y_{\tau,i} - \hat{Y}_{\tau|0,i}\| \leq C_{\tau|0,i}, \forall (\tau, i) \in \{1, \dots, T_{\phi}\} \times \{1, \dots, N\}) \geq 1 - \delta, \quad (3)$$

where  $C_{\tau|0,i}$  indicates the  $\tau$ -step ahead prediction error of agent  $i$  for predictions  $\hat{Y}_{\tau|0,i}$ . In the following, we show how to compute these prediction regions using conformal prediction, and the formal result is given in Theorem 1.

**Data-efficient and accurate prediction regions.** Computing  $\tau$ -step ahead prediction regions via  $C_{\tau|0,i}$  is in general difficult. Existing works (Lindemann et al., 2023a; Stankeviciute, M. Alaa, & van der Schaar, 2021) are conservative, data-inefficient, and do not provide prediction regions for multiple agents. The reason is that  $C_{\tau|0,i}$  are independently computed for all  $(\tau, i) \in \{1, \dots, T_{\phi}\} \times \{1, \dots, N\}$ . This independent computation is followed by a conservative and inefficient union bounding argument (more details below).

Instead, we obtain these prediction regions jointly by extending the framework in Zhao, Hoxha, Fainekos, Deshmukh, and Lindemann (2024) to incorporate multiple agents. For the open-loop case, we define the normalized nonconformity score

$$R_{OL}^{(j)} := \max_{(\tau,i) \in \{1, \dots, T_{\phi}\} \times \{1, \dots, N\}} \frac{\|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|0,i}^{(j)}\|}{\sigma_{\tau|0,i}} \quad (4)$$

<sup>1</sup> Formally, we define the quantile function as  $\text{Quantile}_{1-\delta}(R^{(1)}, \dots, R^{(K)}, \infty) := \inf\{z \in \mathbb{R} | \text{Prob}(Z \leq z) \geq 1 - \delta\}$  with the random variable  $Z := 1/(K + 1)(\sum_i \delta_{R^{(i)}} + \delta_{\infty})$  where  $\delta_{R^{(i)}}$  and  $\delta_{\infty}$  are dirac distributions centered at  $R^{(i)}$  and  $\infty$ , respectively.

for calibration trajectories  $Y^{(j)} \in D_{cal}$  where  $\sigma_{\tau|0,i} > 0$  are constants that normalize the prediction error to  $[0, 1]$  for all times  $\tau$  and agents  $i$ . This nonconformity score is inspired by Cleaveland, Lee, Pappas, and Lindemann (2024), where a mixed integer linear complementarity program is used to find optimal constants  $\sigma_{\tau|0,i}$ . However, we do not need to solve nonconvex optimization problems and instead simply normalize over the training dataset  $D_{train}$  which has practical advantages. Specifically, we use the normalization  $\sigma_{\tau|0,i} := \max_j \|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|0,i}^{(j)}\|$  for training trajectories  $Y^{(j)} \in D_{train}$  with the assumption that  $\sigma_{\tau|0,i}$  is non-zero. This is important as the optimization problem in Cleaveland et al. (2024) may have no solution, or the optimal solution cannot be found due to the nonconvexity of the problem, or it takes too long to solve in practice, especially for large  $T_{\phi}$  and  $N$ . The reader can find an empirical comparison in Section 6.1, where we show that we can obtain similar prediction regions as Cleaveland et al. (2024), but in a computationally more efficient manner.

The intuitions behind the nonconformity score in (4) are that normalization via  $\sigma_{\tau|0,i}$  will prevent the prediction error  $\|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|0,i}^{(j)}\|$  for a specific agent and time to dominate the max operator. This would result in overly conservative prediction regions for other times and agents.

**Valid Prediction Regions with Conformal Prediction.** We can now apply conformal prediction, as introduced in Section 2.2, to the open-loop nonconformity score  $R_{OL}^{(j)}$ . By re-normalizing these nonconformity scores, we obtain prediction regions as in Eqs. (3).

**Theorem 1.** Given the random trajectory  $Y := (Y_0, Y_1, \dots) \sim \mathcal{D}$ , a set of trajectories  $D_{cal}$  and  $D_{train}$  that satisfy Assumption 1, and the failure probability  $\delta \in (0, 1)$ , then the prediction regions in Eq. (3) hold for the choice of

$$C_{\tau|0,i} := C_{OL} \sigma_{\tau|0,i}, \quad (5)$$

where  $\sigma_{\tau|0,i}$  are positive constants and

$$C_{OL} := \text{Quantile}_{1-\delta}(R_{OL}^{(1)}, \dots, R_{OL}^{(K)}, \infty).$$

**Remark 1.** In this work, the prediction regions  $C_{\tau|0,i}$  are norm balls because the nonconformity score is defined using the Euclidean norm. This choice was made for simplicity and interpretability. However, the conformal prediction framework is flexible and can accommodate other nonconformity scores, which would lead to different prediction regions, e.g., ellipsoids (Mesoudi, Destercke, & Rousseau, 2022) or multi-modal regions (Tumu, Cleaveland, Mangharam, Pappas, & Lindemann, 2024). While we chose norm balls for simplicity, the proposed framework is general and can support any convex or even nonconvex shape by defining the nonconformity score appropriately.

**Comparison with existing work.** With slight modification of Lindemann et al. (2023a), Stankeviciute et al. (2021), an alternative approach for obtaining  $C_{\tau|0,i}$  would be to compute  $C_{\tau|0,i} := \text{Quantile}_{1-\delta/(T_{\phi}N)}(R^{(1)}, \dots, R^{(K)}, \infty)$  with the nonconformity score  $R^{(j)} := \|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|0,i}^{(j)}\|$ . However, it is evident that taking the  $1 - \delta$  quantile in Theorem 1 compared to the  $1 - \delta/(T_{\phi}N)$  quantile is much more data inefficient. Specifically, for a fixed  $\delta \in (0, 1)$ , we only require  $K \geq (1 - \delta)/\delta$  calibration trajectories as opposed to  $K \geq (T_{\phi}N - \delta)/\delta$  calibration trajectories.<sup>2</sup> Intuitively, it is also evident that our choice of normalization constants  $\sigma_{\tau|0,i}$  provides less conservative prediction regions in practice as evaluating the  $1 - \delta$  quantile is more favorable compared to the  $1 - \delta/(T_{\phi}N)$

<sup>2</sup> Computing the  $1 - \delta$  and  $1 - \delta/(T_{\phi}N)$  quantiles requires that  $\lceil (K + 1)(1 - \delta) \rceil \leq K$  and  $\lceil (K + 1)(1 - \delta/(T_{\phi}N)) \rceil \leq K$ .

for larger task horizons  $T_\phi$  and number of agents  $N$ . Finally, we remark that one could even consider obtaining conditional guarantees using conditional conformal prediction (Gibbs, Cherian, & Candès, 2023). Similarly, we can use robust conformal prediction to be robust against distribution shifts (Cauchois et al., 2024; Tibshirani et al., 2019; Zhao et al., 2024), e.g., caused by interaction between the system in (1) and the distribution  $\mathcal{D}$ .

**For practitioners.** We summarize our method to obtain the prediction regions in Eq. (3) via Theorem 1 in Algorithm 1, which is executed offline. We note that, in order to compute  $C_{OL}$ , we sort the corresponding nonconformity scores (after adding  $R_{OL}^{(K+1)} := \infty$ ) in nondecreasing order and set  $C_{OL}$  to be the  $p$ th smallest nonconformity score where  $p := \lceil (K+1)(1-\delta) \rceil$ , as explained in Section 2.2. Specifically, the variable  $p$  is set in line 1. We then compute the predictions  $\hat{Y}_{\tau|0,i}^{(j)}$  for all calibration data in lines 2–5 and calculate the normalization constants  $\sigma_{\tau|0,i}$  over the training data in line 6. Finally, we compute the nonconformity scores  $R_{OL}^{(j)}$  for all calibration data in lines 7–9 and apply conformal prediction to obtain  $C_{OL}$  in lines 10–11. According to Theorem 1, we then obtain the open-loop prediction regions  $C_{\tau|0,i}$ .

---

**Algorithm 1:** Multi-Agent Prediction Regions (Offline)

---

**Input:** failure probability  $\delta$ , dataset  $D$ , horizon  $T_\phi$   
**Output:** Constants  $C_{OL}$  and  $\sigma_{\tau|0,i}$

```

1  $p \leftarrow \lceil (K+1)(1-\delta) \rceil$ 
2 forall  $i \in \{1, \dots, N\}$  do
3   forall  $\tau \in [k+1, T_\phi]$  do
4     forall  $j \in \{1, \dots, \bar{K}\}$  do
5       Compute  $\hat{Y}_{\tau|0,i}^{(j)}$  from  $(Y_{0,i}^{(j)}, \dots, Y_{k,i}^{(j)})$ 
6        $\sigma_{\tau|0,i} \leftarrow \max_{j \in \{K+1, \dots, \bar{K}\}} \|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|0,i}^{(j)}\|$ 
7   forall  $j \in \{1, \dots, K\}$  do
8     Compute  $R_{OL}^{(j)}$  as in Eq. (4)
9   Set  $R_{OL}^{(K+1)} \leftarrow \infty$ 
10  Sort  $R_{OL}^{(j)}$  in nondecreasing order
11  Set  $C_{OL} \leftarrow R_{OL}^{(p)}$ 

```

---

#### 4. Open-loop predictive STL control synthesis

We now use the previously computed prediction regions and design predictive controllers that solve Problem 1, i.e., controllers  $u$  that ensure that  $x$  is such that  $\text{Prob}(\mathbf{S} \models \phi) \geq 1-\delta$ . Specifically, we use the prediction regions and present an MIP encoding for the STL task  $\phi$ . We first present a qualitative and a quantitative encoding using the Boolean and quantitative semantics of  $\phi$  in Sections 4.1 and 4.2, respectively. In Section 4.3, we use these encodings to present an open-loop controller.

##### 4.1. Qualitative STL encoding with multi-agent prediction regions

A standard way of encoding STL tasks for deterministic trajectories  $\mathbf{s}$  is by using an MIP encoding (Raman et al., 2014, 2015). The idea is to introduce a binary variable  $z_0^\phi$  and a set of mixed integer constraints over  $\mathbf{s}$  such that  $z_0^\phi = 1$  if and only if  $\mathbf{s} \models \phi$ . In this paper, however, we deal with stochastic trajectories  $\mathbf{S}$  that consist of the system trajectory  $x$  and the stochastic trajectories of dynamic agents  $Y$ . We will instead introduce a binary variable  $\bar{z}_0^\phi$  and a set of mixed integer constraints over  $x$  and the prediction regions in (3) such that  $\bar{z}_0^\phi = 1$  implies that  $\mathbf{S} \models \phi$  with a probability of at least  $1-\delta$ . In this way,  $\bar{z}_0^\phi$  can be seen as a probabilistic version of  $z_0^\phi$ . We emphasize that our MIP encoding provides sufficient but not necessary conditions for the task satisfaction. Specifically, we cannot guarantee necessity due to the use of probabilistic prediction regions.

In the remainder, we generate the mixed integer constraints that define  $\bar{z}_0^\phi$  recursively on the structure of  $\phi$ . Our main innovation is a probabilistic MIP encoding for predicates, while the encoding of Boolean and temporal operators follows (Raman et al., 2014, 2015). We here recall from Assumption 2 that the formula  $\phi$  is in positive normal form so that we do not need to consider negations. To provide a general encoding that can be used for open-loop and closed-loop control, we let  $k \geq 0$  denote the current time. For the open-loop case, we only need to consider the initial time, i.e.,  $k = 0$ . As we will treat the system state  $x_k$  separately from the state of uncontrollable dynamic agents  $Y_k$  contained in  $S_k$ , we will write  $\mu(x_k, Y_k)$  instead of  $\mu(S_k)$ .

**Predicates  $\pi^\mu$ .** Let us denote the set of predicates  $\pi^\mu$  in  $\phi$  by  $\mathcal{P}$ . Now, for each predicate  $\pi^\mu \in \mathcal{P}$  and for each time  $\tau \geq 0$ , we introduce a binary variable  $\bar{z}_{\tau|k}^\mu \in \{0, 1\}$ . If  $\tau \leq k$ , then we have observed the value of  $Y_\tau$  already, and we set  $\bar{z}_{\tau|k}^\mu = 1$  if and only if  $\mu(x_\tau, Y_\tau) \geq 0$ . If  $\tau > k$ , then we would like to constrain  $\bar{z}_{\tau|k}^\mu$  such that  $\bar{z}_{\tau|k}^\mu = 1$  implies  $\text{Prob}(\mu(x_{\tau|k}, Y_\tau) \geq 0) \geq 1-\delta$ . Motivated by Theorem 1, we use the Big-M method and define the constraint

$$-\min_{y \in \mathcal{B}_{\tau|k}} \mu(x_{\tau|k}, y) \leq M(1 - \bar{z}_{\tau|k}^\mu) - \epsilon, \quad (6)$$

where  $M$  and  $\epsilon$  are sufficiently large and sufficiently small positive constants, respectively, see Bemporad and Morari (1999), Raman et al. (2014) for more details. The minimization of  $\mu(x_{\tau|k}, y)$  over the  $y$  component within the set  $\mathcal{B}_{\tau|k}$  will account for all  $Y_\tau$  that are contained within our probabilistic prediction regions. Specifically, the set  $\mathcal{B}_{\tau|k}$  contains all states contained within a geometric norm ball that is centered around the predictions  $\hat{Y}_{\tau|k,i}$  with a size of  $C_{\tau|k,i}$  and is defined as

$$\mathcal{B}_{\tau|k} := \{[y_1, \dots, y_N]^T \in \mathbb{R}^{n_y} \mid \forall i \in \{1, \dots, N\}, \|y_i - \hat{Y}_{\tau|k,i}\| \leq C_{\tau|k,i}\}. \quad (7)$$

In the case when  $k = 0$ , we use the values of  $C_{\tau|0,i}$  as in (5) that define the open-loop prediction region in (3). This implies that  $\text{Prob}(Y_\tau \in \mathcal{B}_\tau, \tau \in \{1, \dots, T_\phi\}) \geq 1-\delta$ .

By Theorem 1 and the construction in Eq. (6), we have the following straightforward result.

**Corollary 1.** If  $\bar{z}_{\tau|0}^\mu = 1$  for all pairs  $(\tau, \pi^\mu) \in TP$ , where  $TP \subseteq \{0, \dots, T_\phi\} \times \mathcal{P}$  is the set of some pairs, then it holds that  $\text{Prob}(\mu(x_{\tau|0}, Y_\tau) \geq 0, \forall (\tau, \pi^\mu) \in TP) \geq 1-\delta$ .

**Remark 2.** We do not need to enforce  $\bar{z}_{\tau|k}^\mu = 0$  since the STL formula  $\phi$  is in positive normal form as per Assumption 2. However, by using the constraint  $\max_{y \in \mathcal{B}_{\tau|k}} \mu(x_{\tau|k}, y) \leq M\bar{z}_{\tau|k}^\mu - \epsilon$  we could enforce  $\bar{z}_{\tau|k}^\mu = 0$  implies  $\text{Prob}(\mu(x_{\tau|k}, Y_\tau) < 0, \forall (\tau, \pi^\mu) \in TP) \geq 1-\delta$ .

**Computational considerations.** Note that we minimize  $\mu(x_{\tau|k}, y)$  over the  $y$  component within the ball  $\mathcal{B}_{\tau|k}$  in Eq. (6). In an outer loop, we will additionally need to optimize over  $x_{\tau|k}$  to compute control inputs  $u_{\tau|k}$ . To obtain computational tractability, we will now remove the minimum over  $y$  and instead write Eq. (6) in closed-form. Since the function  $\mu$  is continuously differentiable and convex in its parameters by Assumption 2, we can use the KKT conditions of the inner problem to do so. Specifically, the constraints in Eq. (6) can be converted into a bilevel optimization problem where the outer problem consists of the constraints

$$-\mu(x_{\tau|k}, y^*) \leq M(1 - \bar{z}_{\tau|k}^\mu) - \epsilon, \quad (8)$$

and where the inner optimization problem is

$$y^* := \underset{y}{\operatorname{argmin}} \mu(x_{\tau|k}, y), \quad (9a)$$

$$\text{s.t. } \|y_i - \hat{Y}_{\tau|k,i}\| \leq C_{\tau|k,i}, \forall i \in \{1, \dots, N\}. \quad (9b)$$

We can use the KKT conditions of the inner optimization problem in (9) after a minor modification in which we change the constraints  $\|y_i - \hat{Y}_{\tau|k,i}\| \leq C_{\tau|k,i}$  into  $(y_i - \hat{Y}_{\tau|k,i})^\top (y_i - \hat{Y}_{\tau|k,i}) \leq C_{\tau|k,i}^2$  to obtain differentiable constraint functions. Formally, we obtain

$$\frac{\partial \mu(x_{\tau|k}, y^*)}{\partial y} + \sum_{i=1}^N \lambda_i \frac{\partial (y_i^* - \hat{Y}_{\tau|k,i})^\top (y_i^* - \hat{Y}_{\tau|k,i})}{\partial y_i} = 0 \quad (10a)$$

$$(y_i^* - \hat{Y}_{\tau|k,i})^\top (y_i^* - \hat{Y}_{\tau|k,i}) \leq C_{\tau|k,i}^2, \forall i \in \{1, \dots, N\} \quad (10b)$$

$$\lambda_i \geq 0, \forall i \in \{1, \dots, N\} \quad (10c)$$

$$\lambda_i ((y_i^* - \hat{Y}_{\tau|k,i})^\top (y_i^* - \hat{Y}_{\tau|k,i}) - C_{\tau|k,i}^2) = 0, \forall i \in \{1, \dots, N\} \quad (10d)$$

as the set of KKT conditions for the inner problem (9).

We have that a solutions  $y^*$  to (9) is a feasible solution to the KKT conditions in (10) and vice versa (Boyd & Vandenberghe, 2004), since Slater's condition holds for the inner convex optimization problem (9), with the reasoning that  $C_{\tau|k,i} > 0$ .<sup>3</sup> As a result, we can replace (6) with Eqs. (8) and (10) equivalently.

**Boolean and temporal operators.** So far, we have encoded predicates  $\pi^\mu$ . We can encode Boolean and temporal operators recursively using the standard MIP encoding (Raman et al., 2014, 2015). In brief, for the conjunction  $\phi := \bigwedge_{i=1}^m \phi_i$  we introduce the binary variable  $\bar{z}_{\tau|k}^\phi \in \{0, 1\}$  that is such that  $\bar{z}_{\tau|k}^\phi = 1$  if and only if  $\bar{z}_{\tau|k}^{\phi_1} = \dots = \bar{z}_{\tau|k}^{\phi_m} = 1$ . Consequently,  $\bar{z}_{\tau|k}^\phi = 1$  implies that  $(\mathbf{S}, \tau) \models \phi$  holds with a probability of at least  $1 - \delta$ . We achieve this by enforcing the constraints

$$\bar{z}_{\tau|k}^\phi \leq \bar{z}_{\tau|k}^{\phi_i}, i = 1, \dots, m,$$

$$\bar{z}_{\tau|k}^\phi \geq 1 - m + \sum_{i=1}^m \bar{z}_{\tau|k}^{\phi_i}.$$

For the disjunction  $\phi := \bigvee_{i=1}^m \phi_i$ , we follow the same idea but instead enforce the constraints

$$\bar{z}_{\tau|k}^\phi \geq \bar{z}_{\tau|k}^{\phi_i}, i = 1, \dots, m,$$

$$\bar{z}_{\tau|k}^\phi \leq \sum_{i=1}^m \bar{z}_{\tau|k}^{\phi_i}.$$

For temporal operators, we follow a similar procedure, but first note that temporal operators can be written as Boolean combinations of conjunctions and disjunctions. For  $\phi := \mathbf{G}_{[a,b]} \phi_1$ , we again introduce a binary variable  $\bar{z}_{\tau|k}^\phi \in \{0, 1\}$  and encode  $\bar{z}_{\tau|k}^\phi = \bigwedge_{\tau'=\tau+a}^{\tau+b} \bar{z}_{\tau'|k}^{\phi_1}$  using the MIP constraints for conjunctions introduced before. Similarly, for  $\phi := \mathbf{F}_{[a,b]} \phi_1$  we use that  $\bar{z}_{\tau|k}^\phi = \bigvee_{\tau'=\tau+a}^{\tau+b} \bar{z}_{\tau'|k}^{\phi_1}$ , while for  $\phi = \phi_1 \mathbf{U}_{[a,b]} \phi_2$  we use that  $\bar{z}_{\tau|k}^\phi = \bigvee_{\tau'=\tau+a}^{\tau+b} (\bar{z}_{\tau'|k}^{\phi_2} \wedge \bigwedge_{\tau''=\tau}^{\tau'} \bar{z}_{\tau''|k}^{\phi_1})$ .

**Soundness of the encoding.** At the initial time step  $k = 0$ , the procedure described above provides us with a binary variable  $\bar{z}_{0|0}^\phi$  and a set of mixed integer constraints over  $x$  such that  $\bar{z}_{0|0}^\phi = 1$  implies that  $\mathbf{S} \models \phi$  holds with a probability of at least  $1 - \delta$ . We summarize this result now and introduce the quantitative encoding thereafter.

**Theorem 2 (Open-Loop Qualitative Encoding Result).** *Let the conditions from Problem 1 hold. If  $\bar{z}_{0|0}^\phi = 1$ , then we have that  $\text{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta$ .*

#### 4.2. Quantitative STL encoding with multi-agent prediction regions

The qualitative MIP encoding ensures that  $\bar{z}_{0|0}^\phi = 1$  implies that  $\text{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta$  as in Theorem 2. However, in some

<sup>3</sup> Here, we make the mild assumption that  $C_{0L} \neq 0$ . Note that  $C_{0L}$  is computed from the nonconformity score  $R^{(j)}$  in (4) which is only zero if our trajectory predictor is perfect, i.e., if  $\|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|0,i}^{(j)}\| = 0$ , which is rarely the case. We could modify the nonconformity score  $R^{(j)}$  by adding a small constant to be guaranteed to achieve  $C_{0L} \neq 0$  at the expense of making the prediction region slightly more conservative.

cases one may want to optimize over the quantitative semantics  $\rho^\phi(\mathbf{S}, 0)$ . We next present a quantitative MIP encoding for  $\phi$ . Specifically, we will recursively define a continuous variable  $\bar{r}_{0|k}^\phi \in \mathbb{R}$  that will be such that  $\rho^\phi(\mathbf{S}, 0) \geq \bar{r}_{0|k}^\phi$  with a probability of at least  $1 - \delta$ . Our main innovation is a quantitative MIP encoding for predicates, while the Boolean and temporal operators again follow the standard MIP encoding (Raman et al., 2014, 2015).

**Predicates.** For each predicate  $\pi^\mu \in \mathcal{P}$  and for each time  $\tau \geq 0$ , we introduce a continuous variable  $\bar{r}_{\tau|k}^\mu \in \mathbb{R}$ . Inspired by Lindemann, Qin, Deshmukh, and Pappas (2023b), we define  $\bar{r}_{\tau|k}^\mu$  as

$$\bar{r}_{\tau|k}^\mu := \begin{cases} \mu(x_\tau, Y_\tau) & \text{if } \tau \leq k, \\ \min_{y \in \mathcal{B}_{\tau|k}} \mu(x_{\tau|k}, y) & \text{otherwise} \end{cases} \quad (11)$$

where we again notice that  $Y_\tau$  is known if  $\tau \leq k$ , while we compute  $\min_{y \in \mathcal{B}_{\tau|k}} \mu(x_{\tau|k}, y)$  if  $\tau > k$  to consider the worst case of  $\mu(x_{\tau|k}, y)$  for  $y$  in the prediction region  $\mathcal{B}_{\tau|k}$ .

By Theorem 1, ensuring that  $Y_\tau \in \mathcal{B}_{\tau|k}$  holds with a probability of at least  $1 - \delta$ , and the construction in Eq. (11), we have the following straightforward result.

**Corollary 2.** *It holds that  $\text{Prob}(\rho^\mu(\mathbf{S}, \tau) \geq \bar{r}_{\tau|0}^\mu, \forall (\tau, \pi^\mu) \in \{0, \dots, T_\phi\} \times \mathcal{P}) \geq 1 - \delta$ , where  $\mathbf{S}_\tau = [x_{\tau|0}, Y_\tau]$ .*

Similarly to the qualitative encoding, the term  $\min_{y \in \mathcal{B}_{\tau|k}} \mu(x_{\tau|k}, y)$  in Eq. (11) can be written as in Eq. (9), and we can use Eq. (10) instead of (9).

**Boolean and temporal operators.** As for the qualitative encoding, we only provide a brief summary for the quantitative encoding of Boolean and temporal operators that follow standard encoding rules. For the conjunction  $\phi := \bigwedge_{i=1}^m \phi_i$ , we introduce the continuous variable  $\bar{r}_{\tau|k}^\phi \in \mathbb{R}$  that will be such that  $\bar{r}_{\tau|k}^\phi = \min\{\bar{r}_{\tau|k}^{\phi_1}, \dots, \bar{r}_{\tau|k}^{\phi_m}\}$ . Particularly, we achieve this by enforcing the constraints

$$\sum_{i=1}^m p_{\tau|k}^{\phi_i} = 1,$$

$$\bar{r}_{\tau|k}^\phi \leq \bar{r}_{\tau|k}^{\phi_i}, i = 1, \dots, m,$$

$$\bar{r}_{\tau|k}^\phi - (1 - p_{\tau|k}^{\phi_i})M \leq \bar{r}_{\tau|k}^\phi \leq \bar{r}_{\tau|k}^{\phi_i} + M(1 - p_{\tau|k}^{\phi_i}), i = 1, \dots, m,$$

where  $p_{\tau|k}^{\phi_i} \in \{0, 1\}$  are  $m$  new binary variables so that  $\bar{r}_{\tau|k}^\phi = r_{\tau|k}^{\phi_i}$  if and only if  $p_{\tau|k}^{\phi_i} = 1$ . By this encoding,  $\bar{r}_{\tau|k}^\phi \geq 0$  implies that  $(\mathbf{S}, \tau) \models \phi$  holds with a probability of at least  $1 - \delta$ . For the disjunction  $\phi := \bigvee_{i=1}^m \phi_i$ , we follow the same idea but instead enforce the constraints

$$\sum_{i=1}^m p_{\tau|k}^{\phi_i} = 1,$$

$$\bar{r}_{\tau|k}^\phi \geq \bar{r}_{\tau|k}^{\phi_i}, i = 1, \dots, m,$$

$$\bar{r}_{\tau|k}^{\phi_i} - (1 - p_{\tau|k}^{\phi_i})M \leq \bar{r}_{\tau|k}^\phi \leq \bar{r}_{\tau|k}^{\phi_i} + M(1 - p_{\tau|k}^{\phi_i}), i = 1, \dots, m.$$

For temporal operators, we again note that we can write each operator as Boolean combinations of conjunctions and disjunctions and then follow the same procedure as for the qualitative encoding.

**Soundness of the encoding.** At the initial time step  $k = 0$ , the procedure described above gives us a continuous variable  $\bar{r}_{0|0}^\phi$  and a set of mixed integer constraints over  $x$  such that  $\bar{r}_{0|0}^\phi$  is a probabilistic lower bound of  $\rho^\phi(\mathbf{S}, 0)$ . We summarize this result next.

**Theorem 3 (Open-Loop Quantitative Encoding Result).** *Let the conditions from Problem 1 hold. It holds that  $\text{Prob}(\rho^\phi(\mathbf{S}, 0) \geq \bar{r}_{0|0}^\phi) \geq 1 - \delta$ . If  $\bar{r}_{0|0}^\phi > 0$ , then we have that  $\text{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta$ .*

We emphasize that the previous result will allow us to directly optimize over the variable  $\bar{r}_{0|0}^\phi$  to achieve  $\bar{r}_{0|0}^\phi > 0$ , which then implies that  $\text{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta$ .



#### 4.3. Predictive MIP-based control synthesis

For control synthesis, we can now use the qualitative encoding via the binary variable  $\bar{z}_{0|k}^\phi$  or the quantitative encoding via the continuous variable  $\bar{r}_{0|k}^\phi$  to solve [Problem 1](#). Formally, we then solve the following qualitative mixed integer optimization problem

$$\mathbf{u}_{k:T_\phi-1}^* := \underset{\mathbf{u}_{k:T_\phi-1}}{\operatorname{argmin}} J(\mathbf{x}_k, \mathbf{u}_{k:T_\phi-1}) \quad (12a)$$

$$\text{subject to} \quad \mathbf{u}_{k:T_\phi-1} \in \mathcal{U}^{T_\phi-k}, \mathbf{x}_{k+1:T_\phi} \in \mathcal{X}^{T_\phi-k} \quad (12b)$$

$$\mathbf{x}_{\tau+1} = f(\mathbf{x}_\tau, \mathbf{u}_\tau), \tau = k, \dots, T_\phi - 1, \quad (12c)$$

$$\bar{z}_{0|k}^\phi = 1. \quad (12d)$$

Similarly, the quantitative mixed integer optimization problem can be formulated by replacing  $\bar{z}_{0|k}^\phi = 1$  with  $\bar{r}_{0|k}^\phi > 0$  in Eq. (12d).

Note particularly that we enforce either  $\bar{z}_{0|k}^\phi = 1$  or  $\bar{r}_{0|k}^\phi > 0$  in Eq. (12d). This enables us to select between the quantitative and the qualitative encoding. We can also enforce  $\bar{r}_{0|k}^\phi > a$ , where  $a > 0$ , to obtain a trajectory with higher robustness.

**Open-loop synthesis.** We will start with analyzing the case when  $k = 0$  where we get an open-loop control sequence  $\mathbf{u}_{0:T_\phi-1}^*$ . By [Theorems 2](#) and [3](#) it is easy to see that we solve [Problem 1](#) if we can find a feasible solution to the optimization problem in (12). We next summarize the result that solves [Problem 1](#).

**Theorem 4 (Open-Loop Control).** *Let the conditions from [Problem 1](#) hold. If the optimization problem in (12) is feasible at time  $k = 0$ , then applying the open-loop control sequence  $\mathbf{u}_{0:T_\phi-1}^*$  to (1) results in*

$$\operatorname{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta.$$

The complexity of the optimization problem in (12) for the open-loop is in general that of a mixed integer program which is NP-hard. Compared to a standard solution to deterministic STL control problems ([Raman et al., 2014](#)), our method introduces additional nonconvexity to the optimization framework. While we maintain identical constraints for Boolean and temporal operators, our formulation extends these constraints by incorporating KKT conditions for each predicate at every time step, which introduces the nonconvex constraint ((10)a) and complementary slackness condition ((10)d).

#### 5. Closed-loop predictive STL control synthesis

The above open-loop control strategy may be conservative as the prediction regions  $C_{\tau|0,i}$  will usually be conservative for large  $\tau$  as then the predictions  $\hat{Y}_{\tau|0}$  lose accuracy. Furthermore, due to the lack of state feedback, the open-loop controller is not robust. Therefore, we propose to solve the optimization problem in (12) in a receding horizon fashion as in [Lindemann et al. \(2023a\)](#).

An intuitive solution to this problem is the MPC framework as follows. At each time  $k$  we observe the realization of  $Y_k$ , update our predictions  $\hat{Y}_{\tau|k}$ , and compute the control sequence  $\mathbf{u}_{k:T_\phi-1}^*$  by solving the optimization problem in (12). Then, we apply the first element  $u_k^*$  of  $\mathbf{u}_{k:T_\phi-1}^*$  to the system in (1) before we repeat the process. However, since the STL task is defined over the whole trajectory  $\mathbf{S}$  (instead of state constraints  $c(\mathbf{x}_k, Y_k) \geq 0$  in [Lindemann et al. \(2023a\)](#)), we cannot apply this strategy and get the same result as in [Lindemann et al. \(2023a\)](#). This poses a new challenge that we address here. Specifically, [Lindemann et al. \(2023a\)](#) show that  $\operatorname{Prob}(c(\mathbf{x}_k, Y_k) \geq 0, \forall k \in \{1, \dots, T\}) \geq 1 - \delta$ , where  $T$  is the length of the task, if the underlying optimization problem is feasible at all time steps. However, this reasoning

cannot be applied to the STL case for the following reason. Our optimization problem (12) depends on both past realizations and future predictions, as opposed to [Lindemann et al. \(2023a\)](#), where the optimization problem only depends on future predictions, but not past realizations. Indeed, past realizations affect the satisfaction of the STL specification and can hence break the feasibility of the optimization problem. The authors in [Stamouli, Lindemann, and Pappas \(2024\)](#) build upon ([Lindemann et al., 2023a](#)) and present a recursively feasible shrinking horizon MPC that guarantees probabilistic satisfaction of state constraints. However, this strategy cannot be applied to the STL case for the same reasons since feasibility is affected by past realizations.

We propose a closed-loop control framework for STL tasks that is recursively feasibility with high probability. We use the qualitative and quantitative encodings presented in Sections 4.1 and 4.2, respectively, and the optimization problem (12) presented in Section 4.3 where we modify the constraints imposed for predicates, while we keep the constraints for Boolean and temporal operators without modification. In order to change the constraints for predicates, we need new prediction regions which we present in Section 5.1 before we discuss the qualitative and quantitative encodings in Section 5.2 as well as the closed-loop controller in Section 5.3.

##### 5.1. Closed-loop prediction regions

We now apply conformal prediction to capture the union of past prediction regions, which will allow us to design a shrinking-horizon MPC with probabilistic recursive feasibility guarantees. Specifically, we are interested in constructing prediction regions for all prediction times  $\tau \in \{1, \dots, T_\phi\}$  made at previous times  $s \in \{0, \dots, \tau - 1\}$  for all agents  $i \in \{1, \dots, N\}$ , i.e., such that

$$\operatorname{Prob}(\|Y_{\tau,i} - \hat{Y}_{\tau|s,i}\| \leq C_{\tau|s,i}, \forall (\tau, s, i) \in \{1, \dots, T_\phi\} \times \{0, \dots, \tau - 1\} \times \{1, \dots, N\}) \geq 1 - \delta, \quad (13)$$

where  $C_{\tau|s,i}$  indicates the prediction error of agent  $i$  for predictions  $\hat{Y}_{\tau|s,i}$  made at time  $s$ . Intuitively, this guarantees that the probability of  $Y_\tau$  being in the previous prediction regions described by  $C_{\tau|s,i}$  for all previous times  $s \in \{0, \dots, \tau - 1\}$  and agents  $i \in \{1, \dots, N\}$  is larger than or equal to  $1 - \delta$ , which will allow us to formulate constraints that guarantee feasibility of the optimization problem (12) (more details follow).

To obtain the prediction region in (13), we follow a similar idea to the open-loop case, but instead define the normalized nonconformity score

$$R_{CL}^{(j)} := \max_{\substack{(\tau, s, i) \in \{1, \dots, T_\phi\} \\ \times \{0, \dots, \tau - 1\} \times \{1, \dots, N\}}} \frac{\|Y_{\tau,i} - \hat{Y}_{\tau|s,i}\|}{\sigma_{\tau|s,i}} \quad (14)$$

for calibration trajectories  $Y^{(j)} \in D_{cal}$  where  $\sigma_{\tau|s,i} > 0$  are again constants that normalize the prediction error to  $[0, 1]$  for all prediction times  $\tau$ , past times  $s$  and agents  $i$ . In this case, we let  $\sigma_{\tau|s,i} := \max_j \|Y_{\tau,i}^{(j)} - \hat{Y}_{\tau|s,i}^{(j)}\|$  for training trajectories  $Y^{(j)} \in D_{train}$ , again assuming that  $\sigma_{\tau|s,i}$  is non-zero. Then, we have the following theorem.

**Theorem 5.** *Given the random trajectory  $Y := (Y_0, Y_1, \dots) \sim \mathcal{D}$ , a set of trajectories  $D_{cal}$  and  $D_{train}$  that satisfy [Assumption 1](#), and the failure probability  $\delta \in (0, 1)$ , then the prediction region in Eq. (13) holds for the choice of*

$$C_{\tau|s,i} := C_{CL} \sigma_{\tau|s,i} \quad (15)$$

where  $\sigma_{\tau|s,i}$  is a positive constant and

$$C_{CL} := \operatorname{Quantile}_{1-\delta}(R_{CL}^{(1)}, \dots, R_{CL}^{(K)}, \infty).$$

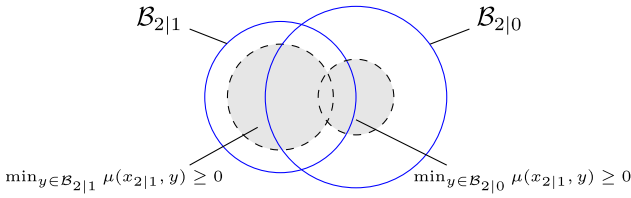


Fig. 1. Illustration of the idea of gradual constraint relaxation.

## 5.2. Closed-loop STL encoding

Future predictions will affect recursive feasibility since the prediction regions made at time  $k + 1$  may be different from that made at time  $k$ , which makes the optimization problems (12) at time  $k + 1$  different from that at time  $k$ . To account for this, inspired by Stamouli et al. (2024), we gradually relax the constraints as now prediction regions become available online. Following this idea, we introduce qualitative and quantitative encodings in the remainder.

**Qualitative encoding.** We replace the quantitative encoding of predicates in Eq. (6) by

$$-\max_{0 \leq s \leq k} \min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) \leq M(1 - \bar{z}_{\tau|k}^{\mu}) - \epsilon, \quad (16)$$

where  $\mathcal{B}_{\tau|s}$  is defined as before in Eq. (7), but now with  $\mathcal{C}_{\tau|s,i}$  as in (15). We use  $s$  to denote the past time steps, and recall that  $k$  is the current time and  $\tau$  is the prediction time. Eq. (16) enforces  $-\min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) \leq M(1 - \bar{z}_{\tau|k}^{\mu}) - \epsilon$  to be satisfied for at least one of the prediction regions  $\mathcal{B}_{\tau|s}$ ,  $0 \leq s \leq k$ , instead of the current prediction region  $\mathcal{B}_{\tau|k}$ . Intuitively, the max operation in (16) allows us to pick the least conservative prediction region to encode the predicate function, which relaxes the constraints gradually. Fig. 1 illustrates the aforementioned idea by an example. Specifically, the predicate describes the task of two agents being close with distance of at most two, i.e.,  $\mu(x, y) = 2 - \|x - y\|$ . The two gray areas indicate the feasible areas at different times, i.e., for the constraints  $\min_{y \in \mathcal{B}_{2|0}} \mu(x, y) \geq 0$  and  $\min_{y \in \mathcal{B}_{2|1}} \mu(x, y) \geq 0$ . The maximum operator in Eq. (16) allows to choose the union of these areas as the feasible set at time step one.

**Computational considerations.** To obtain computational tractability for (16), we introduce a binary variable for each  $s$  as  $q_{\tau|k}^{\mu,s}$  to indicate whether the predicate is satisfied under  $\mathcal{B}_{\tau|s}$ , i.e.,  $q_{\tau|k}^{\mu,s} = 1$  iff  $-\min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) \leq M(1 - \bar{z}_{\tau|k}^{\mu}) - \epsilon$ . Specifically, we have the following constraints for each  $s$ ,  $0 \leq s \leq k$ :

$$-\min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) - M(1 - \bar{z}_{\tau|k}^{\mu}) + \epsilon \leq M'(1 - q_{\tau|k}^{\mu,s}) - \epsilon', \quad (17a)$$

$$\min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) + M(1 - \bar{z}_{\tau|k}^{\mu}) - \epsilon \leq M'q_{\tau|k}^{\mu,s} - \epsilon', \quad (17b)$$

where  $M'$  and  $\epsilon'$  are sufficiently large and sufficiently small positive constants similar to  $M$  and  $\epsilon$ . For simplicity, we can choose  $M' = M$  and  $\epsilon' = \epsilon$ . Then, instead of enforcing (16), we enforce (17) together with

$$\sum_{s=0}^k q_{\tau|k}^{\mu,s} \geq 1. \quad (18)$$

By enforcing (18), it holds that  $q_{\tau|k}^{\mu,s} = 1$  for at least one  $s \in \{0, \dots, k\}$ , so that  $-\min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) \leq M(1 - \bar{z}_{\tau|k}^{\mu}) - \epsilon$  holds for at least one of  $\mathcal{B}_{\tau|s}$ , which implies that (16) holds. In the opposite direction, if  $-\min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) \leq M(1 - \bar{z}_{\tau|k}^{\mu}) - \epsilon$  holds under  $\mathcal{B}_{\tau|s}$ , then  $q_{\tau|k}^{\mu,s} = 1$  and (18) holds. As a result, we can use Eqs. (17) and (18) to replace (16).

**Summary of qualitative encoding.** To encode  $\bar{z}_{0|k}^{\phi}$  in the optimization problem (12), we encode predicates by Eqs. (17) and (18) instead of Eq. (6). Additionally, we encode Boolean and temporal operators in the same way as in the open-loop case in Section 4.1.

**Quantitative encoding.** Similarly, we encode predicates by gradually relaxed constraints for the quantitative encoding. Specifically, we replace the quantitative encoding of predicates in Eq. (11) by

$$\bar{r}_{\tau|k}^{\mu} := \begin{cases} \mu(x_{\tau}, Y_{\tau}) & \text{if } \tau \leq k, \\ \max_{0 \leq s \leq k} \min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) & \text{otherwise.} \end{cases} \quad (19)$$

Intuitively, instead of considering the worst case of  $\mu(x_{\tau|k}, y)$  for  $y$  within the prediction region  $\mathcal{B}_{\tau|k}$ , we choose the least conservative prediction region from  $\mathcal{B}_{\tau|s}$ ,  $0 \leq s \leq k$  to encode the predicate by the maximum operator.

**Computational considerations.** To obtain computational tractability for the maximum operator when  $\tau > k$  in Eq. (19), we apply a similar idea as for the Boolean encoding of the disjunction operator by introducing the binary variable  $q_{\tau|k}^{\mu,s}$  to indicate which value is the largest. Specifically, we use the following constraints to replace (19) in the case that  $\tau > k$ :

$$\sum_{s=0}^k q_{\tau|k}^{\mu,s} = 1, \quad (20a)$$

$$\bar{r}_{\tau|k}^{\mu} \geq \min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y), s = 0, \dots, k, \quad (20b)$$

$$\bar{r}_{\tau|k}^{\phi} \leq \min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) + M(1 - q_{\tau|k}^{\mu,s}), s = 0, \dots, k, \quad (20c)$$

$$\bar{r}_{\tau|k}^{\phi} \geq \min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y) - (1 - q_{\tau|k}^{\mu,s})M, s = 0, \dots, k. \quad (20d)$$

Consequently,  $q_{\tau|k}^{\mu,s} = 1$  iff  $\bar{r}_{\tau|k}^{\mu} = \min_{y \in \mathcal{B}_{\tau|s}} \mu(x_{\tau|k}, y)$ .

**Summary of quantitative encoding.** To encode  $\bar{r}_{0|k}^{\phi}$  in the optimization problem (12), we encode predicates by Eqs. (19) and (20) instead of Eq. (11). We again encode Boolean and temporal operators in the same way as in the closed-loop case in Section 4.2.

The complexity of the optimization problem in (12) for the closed-loop is in general that of a mixed integer program which is NP-hard. Compared to a standard solution to deterministic STL control problems (Raman et al., 2014), our closed-loop formulation at time  $k$  introduces more constraints due to the explicit consideration of all previously predicted regions. This approach generates approximately  $k + 1$  times more nonconvex constraints than that we discussed at the end of Section 4.

## 5.3. Closed-loop predictive control synthesis

We now present our closed-loop control framework. At each time  $k$ , we observe the realization of  $Y_k$ , compute the control sequence  $\mathbf{u}_{k:T_{\phi}-1}^* := u_{k|k}^*, \dots, u_{T_{\phi}-1|k}^*$  based on the previous prediction regions  $\mathcal{B}_{\tau|s}$ ,  $0 \leq s \leq k$ , by solving the optimization problem in (12), where  $\bar{z}_{0|k}^{\phi}$  and  $\bar{r}_{0|k}^{\phi}$  are encoded as we described in the last subsection. Then, we apply the first element  $u_k^*$  of  $\mathbf{u}_{k:T_{\phi}-1}^*$  to the system in (1) before we repeat the process. Naturally, we do so in a shrinking horizon manner where the prediction horizon decreases by one at each time.

**Theorem 6 (Closed-Loop Control).** Let the conditions from Problem 1 hold. Suppose the optimization problem in (12) is feasible at the initial time  $k = 0$ , where the predicates are encoded as in (17) and (18) (qualitative encoding) or (20) (quantitative encoding). Then, the probability of (12) being feasible at every time step  $k \in \{1, \dots, T_{\phi} - 1\}$  is larger than or equal to  $1 - \delta$ . Furthermore, applying  $u_{k|k}^*$  results in  $\text{Prob}(\mathbf{S} \models \phi) \geq 1 - \delta$ .



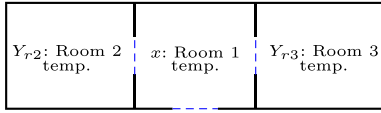
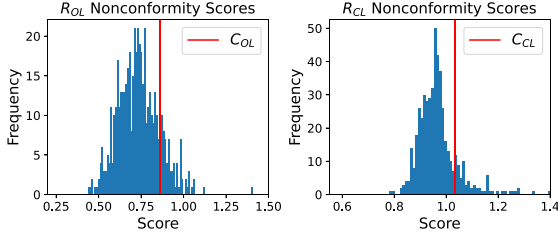
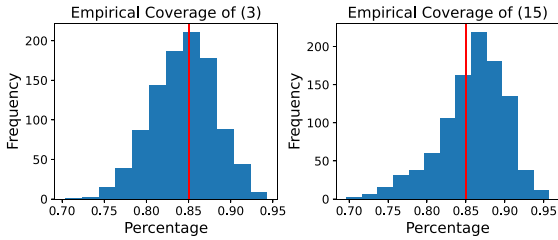


Fig. 2. The floor plan of the hall.

Fig. 3. Nonconformity scores  $R_{OL}^{(j)}$  (left) and  $R_{CL}^{(j)}$  (right) on  $D_{cal}$  in the temperature control case.Fig. 4. Empirical validation results about the coverage of  $R_{OL}^{(j)} \leq C_{OL}$  (left) and  $R_{CL}^{(j)} \leq C_{CL}$  (right) in the temperature control case.

The probabilistic nature of recursive feasibility in our framework arises from the uncertainty in predicting the state  $Y_{k+1}$  of uncontrollable agents at time  $k$ . Since  $Y_{k+1}$  is uncontrollable and unbounded, we cannot provide deterministic guarantees. Instead, we rely on probabilistic statements to ensure that recursive feasibility holds with high probability. Recent literature (Fiacchini, Mammarella, & Dabbene, 2024) has also adopted probabilistic statements to describe recursive feasibility in the presence of unbounded stochastic uncertainty. This probabilistic guarantee is particularly relevant in practical applications where behavior of uncontrollable agents (e.g., pedestrians) cannot be deterministically bounded.

## 6. Case studies

In this section, we illustrate our control method on two case studies. All simulations are conducted in Python 3 and we use SCIP (Bestuzheva et al., 2023) to solve the optimization problem. Our implementations are available at <https://github.com/Xinyi-Yu/STL-Synthesis-among-Uncontrollable-Agents>, where more details can be found.

### 6.1. Building temperature control

We consider the temperature control of a public space in a hall with floor plan shown in Fig. 2. Specifically, we can control the temperature of Room 1, which is the public space, while Rooms 2 and 3 are uncontrollable for us, e.g., these may be meeting rooms where the temperature can only be adjusted manually. Formally, the dynamics of the system are given as  $x_{k+1} = x_k +$

$\tau_s(\alpha_e(T_e - x_k) + \alpha_H(T_h - x_k)u_k)$ , where the state  $x_k \in \mathcal{X} = [0, 45]$  denotes the temperature of the public space at time  $k$ , the control input  $u_k \in \mathcal{U} = [0, 1]$  is the ratio of the heater valve,  $\tau_s$  is the sampling time for the discrete dynamics,  $T_h = 55^\circ\text{C}$  is the heater temperature,  $T_e = 5^\circ\text{C}$  is the outside temperature, and  $\alpha_e = 0.06$  and  $\alpha_H = 0.08$  are heat exchange coefficients. The initial state is  $x_0 = 5$ . The model is adopted from Jagtap et al. (2020).

**System Specification.** The task is to ensure that the difference between the temperatures in the public space and the two meeting rooms is bounded, e.g., such that during transition between rooms people feel comfortable. Let  $Y_{r2}$  and  $Y_{r3}$  denote the uncontrollable random variables describing the temperatures of Rooms 2 and 3. In particular, we require that, the difference between  $x$  and the temperatures of  $Y_{r2}$  and  $Y_{r3}$  should be below a threshold of  $5^\circ\text{C}$  for at least an hour starting from within the first 4 minutes. We set the sampling time to be  $\tau_s = 2$  minutes, and we describe the task as  $\phi := \mathbf{F}_{[0,2]} \mathbf{G}_{[0,30]}(x - Y_{r2} \leq 5 \wedge x - Y_{r2} \geq -5 \wedge x - Y_{r3} \leq 5 \wedge x - Y_{r3} \geq -5)$ . In this case, we set the failure probability to be 15%, i.e.,  $\delta := 0.15$ .

**Behavior of uncontrollable rooms.** The group of people will adjust the temperature in the rooms based on their individual temperature preferences when they enter the meeting room. We assume the temperature variation in  $Y_{r2}$  and  $Y_{r3}$  generated by this adjustment is described by Newton's Law of Cooling (Vollmer, 2009).

**Data Collection.** We collected 2000 trajectories of temperature trajectories for Rooms 2 and 3 – generated as described before. We split them into training (used for the trajectory predictor), calibration (used for conformal prediction), and test (used for validation) datasets with sizes  $|D_{train}| = 500$ ,  $|D_{cal}| = 500$  and  $|D_{test}| = 1000$ , respectively. From the training data  $D_{train}$ , we train a long-short-term memory (LSTM) network to make predictions about the room temperature. We note that we warm-start the LSTM at time  $k = 0$  with data, e.g., the LSTM is fed with past information  $y_{-6}, \dots, y_{-1}$ .

**Multi-agent prediction regions.** Fig. 3 shows histograms of the nonconformity scores  $R_{OL}^{(j)}$  and  $R_{CL}^{(j)}$  evaluated on  $D_{cal}$ . From these histograms of nonconformity scores, we compute  $C_{OL} := 0.863$  and  $C_{CL} := 1.033$  according to Theorems 1 and 5. Next, we empirically validate the correctness of the prediction regions in (3) and (13). We perform the following two experiments 1000 times each. In the first experiment, we randomly sample 150 calibration trajectories from  $D_{cal}$  and 150 test trajectories from  $D_{test}$ . Then, we construct  $C_{OL}$  and  $C_{CL}$  from the 150 calibration trajectories and compute the ratio of how many of the 150 test trajectories satisfy  $R_{OL}^{(j)} \leq C_{OL}$  and  $R_{CL}^{(j)} \leq C_{CL}$ , respectively. Fig. 4 shows the histogram over these ratios, and we can observe that the result achieves the desired coverage  $1 - \delta = 0.85$ . In the second experiment, we randomly sample 150 calibration trajectories from  $D_{cal}$  and 1 test trajectories from  $D_{test}$ . Then, we construct  $C_{OL}$  and  $C_{CL}$  from the 150 calibration trajectories and check whether or not the sampled test data satisfies  $R_{OL}^{(j)} \leq C_{OL}$  and  $R_{CL}^{(j)} \leq C_{CL}$ , respectively. We find a ratio of 0.913 for  $R_{OL}^{(j)} \leq C_{OL}$  and a ratio of 0.871 for  $R_{CL}^{(j)} \leq C_{CL}$ , respectively, which further empirically confirms (3) and (13). For one of these test trajectories, Fig. 5 shows the prediction regions by shaded areas defined by  $\hat{Y}_{\tau|k}$  and  $C_{\tau|k}$  for  $\tau > k$  where  $k = 0$ ,  $k = 8$ ,  $k = 16$  and  $k = 26$  from the left to the right, respectively.

**Prediction region comparison.** We compare our method with Cleaveland et al. (2024), where a parameterized prediction error was proposed. Instead of computing the maximum prediction error  $\sigma$  as in (4) and (14), they proposed an optimization-based method to obtain an in some sense optimal  $\sigma$  that results in small prediction regions (see Cleaveland et al. (2024) for details). Specifically, they formulate the problem as a mixed integer linear complementarity program (MILCP). We extend their framework

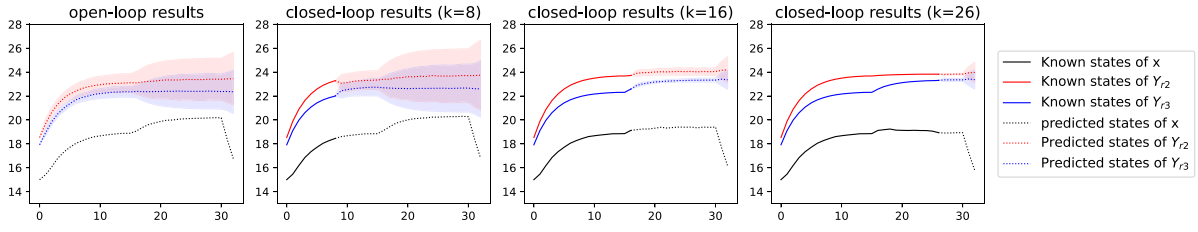


Fig. 5. The result of the temperature qualitative control of the first case study.

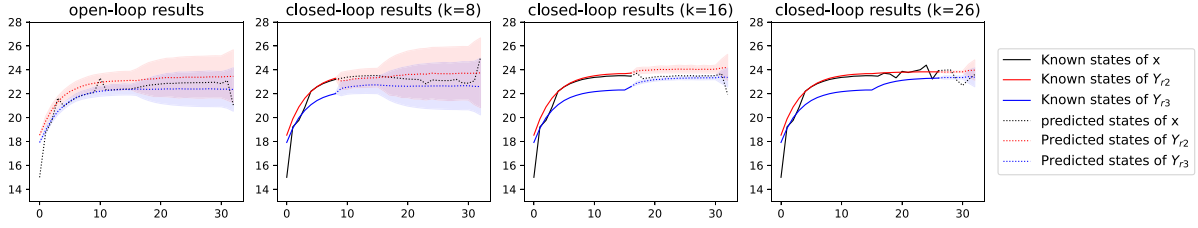


Fig. 6. The result of the temperature quantitative control of the first case study.

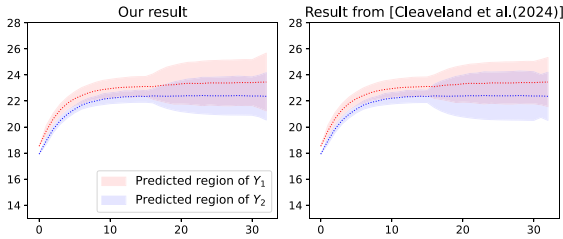


Fig. 7. The comparison result of the open-loop prediction regions of the first case study.

to the multi-agent case and solve the optimization problem with CasADI (Andersson, Gillis, Horn, Rawlings, & Diehl, 2019). The result is shown in Fig. 7. Our computation time is 0.007 s, much less than theirs, which is 2.82 s. Furthermore, the shape of the region looks similar. Note that although they claim they can find the smallest prediction region theoretically, it is sometimes difficult to realize this in practice due to nonconvexity of the optimization problem.

**Qualitative predictive control synthesis.** We run the open-loop and closed-loop control frameworks 1000 times with data from  $D_{test}$ , respectively, where the cost function is set to be the minimum summation of squared control inputs as  $J := \sum_{k=0}^{31} u_k^2$ . In the open-loop control case, the problem is feasible for 1000 times, where the average computation time, average robustness, and task satisfaction rate<sup>4</sup> are 0.148 s, 0.15, and 0.955, respectively. In the closed-loop control case, the problem is feasible at the initial time  $k = 0$  for 999 times,<sup>5</sup> and recursive feasibility is achieved for 986 times, where the average total computation time, average robustness, and task satisfaction rate are 5.90 s, 0.07, and 0.987, respectively. We remark that the closed-loop

<sup>4</sup> The task satisfaction rate is computed as the ratio between the number of times  $\phi$  is satisfied and the number of times the optimization problem is feasible at time  $k = 0$ .

<sup>5</sup> We note the difference in initial feasibility compared to the open-loop controller, which is caused since prediction regions  $\mathcal{B}_{r|0}$  of the closed-loop controller are more conservative.

controller has slightly lower robustness than the open-loop controller. This is since the closed-loop prediction regions  $\mathcal{B}_{r|k}$  for large  $k$  appear to be less conservative than open-loop prediction regions. We also see that the closed-loop controller has a higher task satisfaction rate since the controller can update the control inputs in a receding horizon fashion to observations of the uncontrollable agents. Fig. 5 shows one of the results of the proposed control framework with the qualitative encoding. Fig. 5(a) specifically presents the result of the open loop controller where the prediction regions are large for large  $\tau$ , while Figs. 5(b)(c)(d) present the result of the closed loop controller at times  $k = 8, 16, 26$ , respectively.

**Quantitative predictive control synthesis.** Similar to the qualitative control, we run the open-loop and closed-loop control frameworks with quantitative encoding 1000 times with data in  $D_{test}$ , respectively, where we enforce the robust semantics to be larger than 1, i.e.,  $\bar{r}_{0|k}^\phi \geq 0$ , and we maximize  $\bar{r}_{0|k}^\phi$  in the cost function, i.e., we set  $J := -\bar{r}_{0|k}^\phi$ . In the open-loop control case, the problem is feasible for 927 times, where the average computation time, average robustness, and task satisfaction rate are 1.79 s, 2.92, and 1, respectively. In the closed-loop control case, the problem is feasible at the initial time  $k = 0$  for 923 times, and the recursive feasibility is achieved for 890 times, where the average computation time, average robustness, and task satisfaction rate are 147.29 s, 2.80, and 0.964, respectively. The closed-loop controller has a lower task satisfaction rate than the open-loop controller due to the nonconvexity of the optimization problem and performance issues of our solver in practice. It is worthy noting that the computation time of the quantitative encoding is greater than that of the qualitative encoding, although it may achieve better robustness performance. As we can see in the result in Fig. 6, the trajectory stays near the center of the prediction regions, leading to higher robustness.

## 6.2. Robot motion planning

We consider the planar motion of a single robot (denoted by  $R_1$ ) with double integrator dynamics and a sampling period of 1 s,

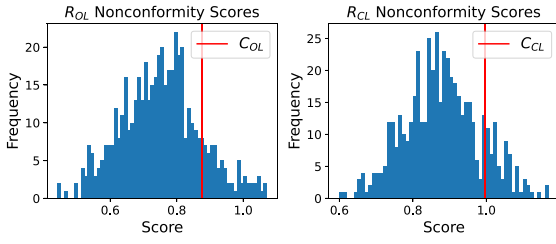


Fig. 8. Nonconformity scores  $R_{OL}^{(j)}$  (left) and  $R_{CL}^{(j)}$  (right) on  $D_{cal}$  in the robot motion planning case.

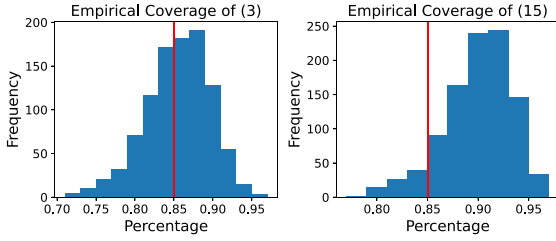


Fig. 9. Empirical validation results about the coverage of  $R_{OL}^{(j)} \leq C_{OL}$  (left) and  $R_{CL}^{(j)} \leq C_{CL}$  (right) in the robot motion planning case.

which is described as

$$x_{k+1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix} u_k$$

where the state  $x_k = [p_x^1 \ v_x^1 \ p_y^1 \ v_y^1]^\top$  denotes the  $x$ -position,  $x$ -velocity,  $y$ -position and  $y$ -velocity, and where the control input  $u_k = [u_x \ u_y]^\top$  denotes the  $x$ -acceleration and  $y$ -acceleration, respectively. The physical constraints are  $x \in \mathcal{X} = [0, 10] \times [-1.5, 1.5] \times [0, 10] \times [-1.5, 1.5]$  and  $u \in \mathcal{U} = [-1, 1] \times [-1, 1]$ . The initial state is  $x_0 = [1, 0, 1, 0]^\top$ . Furthermore, we have one more uncontrollable agent (denoted by  $R_2$ ) and its state  $Y_k = [p_x^2 \ p_y^2]^\top$  denotes the  $x$ -position and  $y$ -position. Then we define the joint state as  $s_k = [p_x^1 \ v_x^1 \ p_y^1 \ v_y^1 \ p_x^2 \ p_y^2]^\top$ . In this case,  $R_1$  is controllable and  $R_2$  is uncontrollable.

**System Specification.** The objective of the two robots is to visit specific regions to perform specific tasks, such as collecting packages and transporting them to a final destination. For this collaborative task,  $R_2$  is the leader and  $R_1$  should follow  $R_2$ . Specifically,  $R_2$  should arrive and stay in the left blue region and top blue region, as shown in Fig. 10, within the time intervals  $[4, 6]$  and  $[9, 13]$ , respectively. After that, starting from a certain time instant between the time interval  $[16, 18]$ ,  $R_1$  and  $R_2$  should stay in bottom right red and blue regions respectively for at least 2 time instants. During the task execution, the distances between two agents should be at most  $D$  meters such that two robots can communicate or perform collaborative tasks. Furthermore, two robots should always avoid the obstacles (gray shaded areas). In particular, such a task can be described by:

$$\phi := \mathbf{G}_{[4,6]} \pi^{\mu_1} \wedge \mathbf{G}_{[9,13]} \pi^{\mu_2} \wedge \mathbf{F}_{[16,18]} \mathbf{G}_{[0,2]} (\pi^{\mu_3} \wedge \pi^{\mu_4}) \\ \wedge \mathbf{G}_{[0,20]} (\pi^{\mu_{close}} \wedge \pi^{\mu_{obs1}} \wedge \pi^{\mu_{obs2}}),$$

where  $\mu_1 := \min(p_x^2, 2 - p_x^2, p_y^2 - 4, 6 - p_y^2)$ ,  $\mu_2 := \min(p_x^2 - 3.5, 6.5 - p_x^2, p_y^2 - 8, 10 - p_y^2)$ ,  $\mu_3 := \min(p_x^2 - 8.5, 10 - p_x^2, p_y^2 - 2 - p_y^2)$  and  $\mu_4 := \min(p_x^1 - 7, 8.5 - p_x^1, p_y^1 - 2 - p_y^1)$  represent the tasks for  $R_2$  in the left blue region,  $R_2$  in the top blue region,  $R_2$  in the bottom right blue region, and  $R_1$  in the bottom red

region, respectively. Here,  $\mu_{close} := D - \min(p_x^2 - p_x^1, p_x^1 - p_x^2, p_y^2 - p_y^1, p_y^1 - p_y^2)$  is the predicate regarding the distance requirement between the two robots where we set  $D = 2$ . The predicates  $\mu_{obs1} := \min(\max(1.6 - p_x^1, p_x^1 - 2.6, 2 - p_y^1, p_y^1 - 3), \max(8.3 - p_x^1, p_x^1 - 9.3, 6.5 - p_y^1, p_y^1 - 7.5), \max(5.7 - p_x^1, p_x^1 - 6.7, 2.7 - p_y^1, p_y^1 - 3.7))$  and  $\mu_{obs2} := \min(\max(1.6 - p_x^2, p_x^2 - 2.6, 2 - p_y^2, p_y^2 - 3), \max(8.3 - p_x^2, p_x^2 - 9.3, 6.5 - p_y^2, p_y^2 - 7.5), \max(5.7 - p_x^2, p_x^2 - 6.7, 2.7 - p_y^2, p_y^2 - 3.7))$  describe the tasks for obstacle avoidance. We decompose  $\phi$  into two task for each of the two robots as follows

$$\phi_{R_1} := \mathbf{F}_{[16,18]} \mathbf{G}_{[0,2]} \pi^{\mu_4} \wedge \mathbf{G}_{[0,20]} (\pi^{\mu_{close}} \wedge \pi^{\mu_{obs1}}),$$

$$\phi_{R_2} := \mathbf{G}_{[4,6]} \pi^{\mu_1} \wedge \mathbf{G}_{[9,13]} \pi^{\mu_2} \wedge \mathbf{F}_{[16,18]} \mathbf{G}_{[0,2]} \pi^{\mu_3} \wedge \mathbf{G}_{[0,20]} \pi^{\mu_{obs2}}.$$

As  $R_2$  will lead the task, its behavior will not be influenced by  $R_1$ . On the other hand,  $R_1$  is responsible for the task  $\mu_{close}$  which means it should always track the behavior of  $R_2$ . Regarding the execution of the task, we only enforce  $R_1$  to achieve  $\phi_{R_1}$  instead of  $\phi$ , which makes sense in the leader-follower setting of this case study.

**Motion of uncontrollable agent.** We set  $R_2$ 's dynamical system as  $y_{k+1} = f(y_k, u_k^y) + \omega_k$  where  $f(y_k, u_k^y)$  describes the same double integrator dynamics as for  $R_1$ , and  $\omega_k \in \mathcal{W}$  is a uniformly distributed disturbance from the set  $\mathcal{W} = ([-0.15, 0.15] \times [0, 0])^2$ . We compute an closed-loop control sequence for  $R_2$  under the task  $\phi_{R_2}$  using a standard MIP encoding (Raman et al., 2014).

**Data collection.** We collected 2000 trajectories of the uncontrollable agent. We split the data into training, calibration, and test datasets with sizes  $|D_{train}| = 500$ ,  $|D_{cal}| = 500$ , and  $|D_{test}| = 1000$ , respectively. We again trained an LSTM on  $D_{train}$  for trajectory prediction.

**Prediction regions.** Fig. 8 shows histograms of the nonconformity scores  $R_{OL}^{(j)}$  and  $R_{CL}^{(j)}$  evaluated on  $D_{cal}$ . Based on these nonconformity scores, we have that  $C_{OL} = 0.876$  and  $C_{CL} = 0.997$  by using  $\delta = 0.15$ . Next, we empirically validate the correctness of the prediction regions by checking whether or not (3) and (13) hold on  $D_{test}$  by the same two experiments as the temperature control case. Fig. 9 shows the histogram over the result of the ratios in the first experiment, and we can observe that the result achieves the desired coverage  $1 - \delta = 0.85$ . In the second experiment, we obtain a ratio of 0.854 for  $R_{OL}^{(j)} \leq C_{OL}$  and a ratio of 0.894 for  $R_{CL}^{(j)} \leq C_{CL}$ , respectively, which also empirically confirms (3) and (13). For one of these test trajectories, Fig. 10 shows the prediction regions defined by  $\hat{Y}_{\tau|k}$  and  $C_{\tau|k}$  for  $\tau > k$  where  $k = 0$ ,  $k = 6$ ,  $k = 12$  and  $k = 18$ , respectively.

**Predictive control synthesis.** As in our first experiment, we run the open-loop and the closed-loop controller with the qualitative encoding 1000 times with data from  $D_{test}$ . The cost function encodes the trade off between the sum of the squared control inputs and velocities as  $J := \sum_{k=0}^{20} (0.97 \|u_k\|^2 + 0.03 (v_x^1)^2 + 0.03 (v_y^1)^2)$ . In the open-loop control case, the problem is feasible for 1000 times, where the average computation time, average robustness, and task satisfaction rate are 0.70 s, 0.06490, and 1, respectively. In the closed-loop control case, the problem is feasible at the initial time  $k = 0$  for 1000 times, with recursive feasibility in all cases, where the average total computation time, average robustness, and task satisfaction rate are 14.80 s, 0.06489, and 1, respectively. The closed-loop case has again a slightly lower robustness for the same reasons as in the first experiment. Fig. 10 shows one of the results of the proposed control framework, where we can see that robot  $R_1$  will use the prediction regions of  $R_2$  to compute a control input that results in task satisfaction. Fig. 10(a) specifically presents the result of the open loop controller while Figs. 10(b)(c)(d) present the result of the closed loop controller at times  $k = 6, 12, 18$ , respectively.



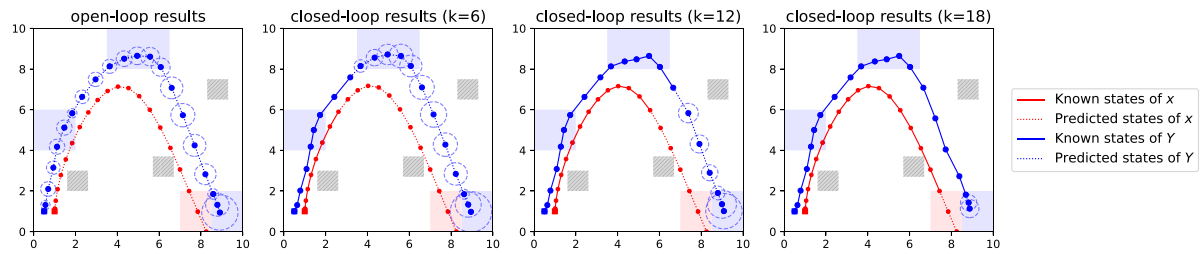


Fig. 10. The result of the robot motion planning of the second case study.

## References

- Akella, Prithvi, Dixit, Anushri, Ahmadi, Mohamadreza, Burdick, Joel W, & Ames, Aaron D (2024). Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification. *Artificial Intelligence*.  
 Andersson, J., Gillis, J., Horn, G., Rawlings, J., & Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.  
 Angelopoulos, A., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.  
 Ayala, A., Andersson, S., & Belta, C. (2011). Temporal logic control in dynamic environments with probabilistic satisfaction guarantees. In *IEEE/RSJ IROS*.  
 Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*.  
 Bestuzheva, K., Besançon, M., Chen, W., Chmiela, A., Donkiewicz, T., van Doornmalen, J., et al. (2023). Enabling research through the SCIP optimization suite 8.0. *Association for Computing Machinery. Transactions on Mathematical Software*.  
 Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*.  
 Büyükoçak, A., Aksaray, D., & Yazicioglu, Y. (2021). Distributed planning of multi-agent systems with coupled temporal logic specifications. In *AIAA sciTech 2021 forum*.  
 Cauchois, M., Gupta, S., Ali, A., & Duchi, J. (2024). Robust validation: Confident predictions even when distributions shift. *Journal of the American Statistical Association*.  
 Charitidou, M., & Dimarogonas, D. (2022). Receding horizon control with online barrier function design under signal temporal logic specifications. *IEEE TAC*.  
 Cleaveland, M., Lee, I., Pappas, G., & Lindemann, L. (2024). Conformal prediction regions for time series using linear complementarity programming. In *AAAI*.  
 Dixit, A., Lindemann, L., Wei, S., Cleaveland, M., Pappas, G., & Burdick, J. (2023). Adaptive conformal prediction for motion planning among dynamic agents. In *L4DC. PMLR*.  
 Dokhanchi, A., Hoxha, B., & Fainekos, G. (2014). On-line monitoring for temporal logic robustness. In *RV*.  
 Donzé, A., & Maler, O. (2010a). Robust satisfaction of temporal logic over real-valued signals. In *FORMATS*.  
 Donzé, A., & Maler, O. (2010b). Robust satisfaction of temporal logic over real-valued signals. In *Formal modeling and analysis of timed systems*.  
 Fainekos, G., & Pappas, G. (2009). Robustness of temporal logic specifications for continuous-time signals. *TCS*.  
 Farahani, S., Majumdar, R., Prabhu, V., & Soudjani, S. (2017). Shrinking horizon model predictive control with chance-constrained signal temporal logic specifications. In *ACC*.  
 Fiacchini, M., Mammarella, M., & Dabbene, F. (2024). Measured-state conditioned recursive feasibility for stochastic model predictive control. *arXiv preprint arXiv:2406.13522*.  
 Fontana, M., Zeni, G., & Vantini, S. (2023). Conformal prediction: a unified review of theory and new challenges. *Bernoulli*.  
 Gibbs, I., & Candes, E. (2021). Adaptive conformal inference under distribution shift. *NeurIPS*.  
 Gibbs, I., Cherian, J., & Candès, E. (2023). Conformal prediction with conditional guarantees. *arXiv preprint arXiv:2305.12616*.  
 Gilpin, Y., Kurtz, V., & Lin, H. (2020). A smooth robustness measure of signal temporal logic for symbolic control. *IEEE L-CSS*.  
 Gundana, D., & Kress-Gazit, H. (2021). Event-based signal temporal logic synthesis for single and multi-robot tasks. *IEEE RAL*.  
 Gundana, D., & Kress-Gazit, H. (2022). Event-based signal temporal logic tasks: Execution and feedback in complex environments. *IEEE RAL*.  
 Guo, M., & Dimarogonas, D. (2015). Multi-agent plan reconfiguration under local LTL specifications. *IJRR*.  
 Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. *NeurIPS*.  
 Hashemi, N., Qin, X., Deshmukh, J., Fainekos, G., Hoxha, B., Prokhorov, D., et al. (2023). Risk-awareness in learning neural controllers for temporal logic objectives. In *ACC*.  
 Hoxha, B., & Fainekos, G. (2016). Planning in dynamic environments through temporal logic monitoring. In *AAAI workshop*.  
 Ilyes, R., Ho, Q., & Lahijanian, M. (2023). Stochastic robustness interval for motion planning with signal temporal logic. In *IEEE iCRA*.  
 Jagtap, P., Soudjani, S., & Zamani, M. (2020). Formal synthesis of stochastic systems via control barrier certificates. *IEEE TAC*.  
 Jha, S., Raman, V., Sadigh, D., & Seshia, S. (2018). Safe autonomy under perception uncertainty using chance-constrained temporal logic. *Journal of Automated Reasoning*.  
 Kalluraya, S., Pappas, G., & Kantaros, Y. (2023). Multi-robot mission planning in dynamic semantic environments. In *IEEE iCRA*.  
 Kantaros, Y., Guo, M., & Zavlanos, M. (2019). Temporal logic task planning and intermittent connectivity control of mobile robot networks. *IEEE TAC*.  
 Kantaros, Y., & Zavlanos, M. (2020). Stylus: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems. *IJRR*.  
 Kloetzer, M., & Belta, C. (2009). Automatic deployment of distributed teams of robots from temporal logic motion specifications. *T-RO*.  
 Kloetzer, M., & Mahulea, C. (2012). LTL planning in dynamic environments. *IFAC Proceedings Volumes*.  
 Kurtz, V., & Lin, H. (2022). Mixed-integer programming for signal temporal logic with fewer binary variables. *IEEE L-CSS*.  
 Li, Y., Shahrivar, E., & Liu, J. (2021). Safe linear temporal logic motion planning in dynamic environments. *IEEE/RSJ IROS*.  
 Lindemann, L., Cleaveland, M., Shim, G., & Pappas, G. (2023a). Safe planning in dynamic environments using conformal prediction. *IEEE RAL*.  
 Lindemann, L., & Dimarogonas, D. (2018). Control barrier functions for signal temporal logic tasks. *IEEE L-CSS*.  
 Lindemann, L., & Dimarogonas, D. (2019). Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks. *IEEE L-CSS*.  
 Lindemann, L., Pappas, G., & Dimarogonas, D. (2021). Reactive and risk-aware control for signal temporal logic. *IEEE TAC*.  
 Lindemann, L., Qin, X., Deshmukh, J., V, & Pappas, G. (2023b). Conformal prediction for STL runtime verification. *ICCPs*.  
 Maler, O., & Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In *FORMAT*.  
 Mehdipour, N., Vasile, C., & Belta, C. (2019). Arithmetic-geometric mean robustness for control from signal temporal logic specifications. In *ACC*.  
 Messoudi, S., Destercke, S., & Rousseau, S. (2022). Ellipsoidal conformal inference for multi-target regression. In *Conformal and probabilistic prediction with applications. PMLR*.  
 Muthali, A., Shen, H., Deglurkar, S., Lim, M., Roelofs, R., Faust, A., et al. (2023). Multi-agent reachability calibration with conformal prediction. In *IEEE CDC. IEEE*.  
 Pant, Y., Vardhan, A., Houssam, & Mangharam, R. (2017). Smooth operator: Control using the smooth robustness of temporal logic. In *IEEE cCTA*.  
 Pant, Y., Abbas, H., Quaye, R., & Mangharam, R. (2018). Fly-by-logic: Control of multi-drone fleets with temporal logic objectives. In *ICCPs*.  
 Purohit, P., & Saha, I. (2021). DT: Temporal logic path planning in a dynamic environment. In *IEEE/RSJ IROS*.  
 Raman, V., Donzé, A., Maasoumy, M., Murray, R., Sangiovanni-Vincentelli, A., & Seshia, S. (2014). Model predictive control with signal temporal logic specifications. In *IEEE CDC*.

- Raman, V., Donzé, A., Sadigh, D., Murray, R., & Seshia, S. (2015). Reactive synthesis from signal temporal logic specifications. In *HSCC*.
- Sadigh, D., & Kapoor, A. (2016). Safe control under uncertainty with probabilistic signal temporal logic. In *RSS*.
- Sadraddini, S., & Belta, C. (2015). Robust temporal logic model predictive control. In *Allerton*.
- Sadraddini, S., & Belta, C. (2018). Formal synthesis of control strategies for positive monotone systems. *IEEE TAC*.
- Safaoui, S., Lindemann, L., Dimarogonas, D., Shames, I., & Summers, T. (2020). Control design for risk-based signal temporal logic specifications. *IEEE L-CSS*.
- Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint [arXiv:1801.01078](https://arxiv.org/abs/1801.01078).
- Scher, G., Sadraddini, S., Yadin, A., & Kress-Gazit, H. (2023). Ensuring reliable robot task performance through probabilistic rare-event verification and synthesis. arXiv preprint [arXiv:2304.14886](https://arxiv.org/abs/2304.14886).
- Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *JMLR*.
- Srinivasan, M., Coogan, S., & Egerstedt, M. (2018). Control of multi-agent systems with finite time control barrier certificates and temporal logic. In *IEEE CDC*.
- Stamouli, C., Lindemann, L., & Pappas, G. (2024). Recursively feasible shrinking-horizon MPC in dynamic environments with conformal prediction guarantees. In *L4DC*. PMLR.
- Stankeviciute, K., M. Alaa, A., & van der Schaar, M. (2021). Conformal time-series forecasting. *NeurIPS*.
- Sun, D., Chen, J., Mitra, S., & Fan, C. (2022). Multi-agent motion planning from signal temporal logic specifications. *IEEE RAL*.
- Tibshirani, R., Foygel Barber, R., Candès, E., & Ramdas, A. (2019). Conformal prediction under covariate shift. *NeurIPS*.
- Tonkens, S., Sun, S., Yu, R., & Herbert, S. (2023). Scalable safe long-horizon planning in dynamic environments leveraging conformal prediction and temporal correlations.
- Tumu, R., Cleaveland, M., Mangharam, R., Pappas, G., & Lindemann, L. (2024). Multi-modal conformal prediction regions by optimizing convex shape templates. In *L4DC*. PMLR.
- Ulusoy, A., & Belta, C. (2014). Receding horizon temporal logic control in dynamic environments. *IJRR*.
- Vasile, C., Raman, V., & Karaman, S. (2017). Sampling-based synthesis of maximally-satisfying controllers for temporal logic specifications. In *IROS*.
- Vollmer, M. (2009). Newton's law of cooling revisited. *European Journal of Physics*.
- Vovk, V., Gammernan, A., & Shafer, G. (2005). Algorithmic learning in a random world.
- Xiao, W., Belta, C., & Cassandras, C. (2021). High order control lyapunov-barrier functions for temporal logic specifications. In *ACC*.
- Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*.
- Yu, X., Zhao, Y., Yin, X., & Lindemann, L. (2023). Signal temporal logic control synthesis among uncontrollable dynamic agents with conformal prediction. arXiv preprint [arXiv:2312.04242](https://arxiv.org/abs/2312.04242).
- Zhao, Y., Hoxha, B., Fainekos, G., Deshmukh, J., & Lindemann, L. (2024). Robust conformal prediction for stl runtime verification under distribution shift. In *ICCPs*. IEEE.

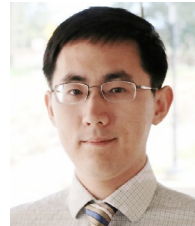


**Xinyi Yu** was born in Jiangsu, China, in 1999. She received the B.Eng degree from China University of Petroleum (East China) in Qingdao in 2020, and the M.S. degree from Shanghai Jiao Tong University in Shanghai in 2023, both in automation. She is pursuing the Ph.D. degree in computer science at University of Southern California.



**Yiqi Zhao** received his B.S. degree with Honors in Computer Science from Vanderbilt University, Nashville, TN, USA in 2023. He also completed a secondary major in Mathematics and minors in Electrical Engineering and Data Science at Vanderbilt University. He received his M.S. degree in Computer Science from the University of Southern California (USC), Los Angeles, CA, USA in 2025 and is currently working towards the Ph.D. degree in Computer Science from USC. His research interests include robotics, formal methods, cyber-physical systems, and systems and control theory. He was a finalist

for the Best Paper Award at the 2024 International Conference on Cyber-Physical Systems.



**Xiang Yin** was born in Anhui, China, in 1991. He received the B.Eng degree from Zhejiang University in 2012, the M.S. degree from the University of Michigan, Ann Arbor, in 2013, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 2017, all in Electrical Engineering. Since 2017, he has been with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, where he is a Full Professor. His research interests include formal methods, discrete-event systems and cyber-physical systems.

Dr. Yin is serving as the chair of the *IEEE CSS Technical Committee on Discrete Event Systems*, Associate Editors for the *Journal of Discrete Event Dynamic Systems: Theory & Applications*, *IEEE Control Systems Letters*, *Nonlinear Analysis: Hybrid Systems*, and a member of the *IEEE CSS Conference Editorial Board*.



**Lars Lindemann** is an Assistant Professor for Algorithmic Systems Theory in the Automatic Control Laboratory at ETH Zürich. He received his Ph.D. degree in Electrical Engineering from KTH Royal Institute of Technology in 2020, was a Postdoctoral Fellow in the Department of Electrical and Systems Engineering at the University of Pennsylvania from 2020 to 2022, and an Assistant Professor in the Department of Computer Science at the University of Southern California from 2023 to 2025. His research interests include systems and control theory, formal methods, machine learning,

and autonomous systems.