Safe Learning MPC With Limited Model Knowledge and Data

Aaron Kandel and Scott J. Moura

Abstract— **This article presents an end-to-end framework for safe learning-based control (LbC) using nonlinear stochastic MPC and distributionally robust optimization (DRO). This work is motivated by several open challenges in LbC literature. Many control-theoretic LbC methods require subject matter expertise (SME), often manifested as preexisting data of safe trajectories or structural model knowledge, to translate their own safety guarantees. In this article, we focus on LbC where the controller is applied directly to a system of which it has no or extremely limited direct experience, toward safety during *tabula-rasa* or "*blank slate*" model-based learning and control as a challenging case for validation. This explores the boundary of the status quo in control theory relating to requirements for SME. We show under basic and limited assumptions on the underlying problem, we can translate probabilistic guarantees on the feasibility of nonlinear systems using results in stochastic MPC and DRO literature whose relevance we formally extend in mathematical analysis. We also present a coupled and intuitive formulation for the persistence of excitation (PoE) and illustrate the connection between PoE and the applicability of the proposed method. Our case studies of vehicle obstacle avoidance and safe extremely fast charging of lithium-ion batteries reveal powerful empirical results supporting the underlying theory.**

Index Terms— **Adaptive control, data-driven control, energy systems, learning, lithium-ion battery, model-predictive control, robust optimization, vehicle autonomy.**

I. Introduction

THIS article presents a novel application of Wasserstein ambiguity sets to robustify model-based reinforcement learning (MBRL) and learning-based control (LbC) in safety-critical applications. Here, we define safety as the ability of the control policy to satisfy constraints. Translating safety to online reinforcement learning (RL) algorithms is a notoriously difficult open challenge in relevant literature. This article is motivated by unsolved shortcomings of many existing means to address this challenge, particularly a strong and often optimistic dependence on subject matter expertise (SME). Two overarching examples include: 1) assumed knowledge of underlying dynamics and 2) preexisting data of safe trajectories.

The LbC problem space borrows many concepts from historical research on stochastic optimal control, a field which dates back decades to the original linear–quadratic Gaussian problem [1]. The key underlying concept relates to uncertainty, and how we can accommodate limited or imperfect knowledge of the underlying dynamics. The rise in popularity of MPC has created a new application for these robust and stochastic control principles. For instance, foundational work by Kothare et al. addresses uncertainty in MPC optimization with linear matrix inequalities by allowing the state transition matrices to vary in time within a convex polytope [2].

Within the past few years, stochastic optimal control has become connected to ongoing research in the burgeoning field of LbC. Here, researchers seek guarantees of safety and performance when learning-based controlling a dynamical system simultaneously. For a review of current state-of-the-art methods in LbC that utilize MPC, we direct the reader to a thorough review by Hewing et al. [3]. This type of problem presents a nuanced and complex challenge for a host of reasons. Safety and feasibility pose significant barriers to the proper implementation of such algorithms. Moreover, balancing the exploration-exploitation tradeoff inherent to simultaneous control and model identification has presented researchers with a host of unique problems that form a primary focus of research in active learning. Dean et al. [4], for instance, explores the safety and persistence of excitation (PoE) for a learned constrained linear–quadratic regulator.

MPC is a highly popular use case for LbC problems and provides an intuitive bridge between longstanding adaptive control theory and new developments and explorations. For instance, recent work has investigated recursive feasibility for adaptive MPC controllers based on recursive least-squares [5] and set-membership parameter identification [6], although similar articles frequently possess limitations including dependence on linear dynamical models. Rosolia and Borrelli derive recursive feasibility and performance guarantees for a learned episodic MPC controller [7]. Koller et al. [8] also addressed the safety of a learned MPC controller when imperfect model knowledge and safe control exist.

We note that control Lyapunov function and control barrier function [9], [10], [11]-based approaches have further strengthened the connection between classical adaptive control and more modern approaches akin to popular model-based

Manuscript received 18 May 2022; revised 24 January 2023 and 16 August 2023; accepted 6 October 2023. This work was supported by the National Science Foundation Graduate Research Fellowship. Recommended by Associate Editor K. Barton. *(Corresponding author: Aaron Kandel.)*

Aaron Kandel is with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94709 USA (e-mail: aaronkandel@berkeley.edu).

Scott J. Moura is with the Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, CA 94709 USA (e-mail: smoura@berkeley.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2023.3324869.

Digital Object Identifier 10.1109/TCST.2023.3324869

RL problems. Recent work by Westenbrouk et al. [12] has even explored coupling such nonlinear control methods with a policy optimization scheme.

In the space of RL, safe LbC has become a burgeoning area of study. For broad discussion and categorization of classical methods, Garcia and Fernández [13] provided a comprehensive review. More recently, some control-theoretic principles have migrated toward the space of safe RL. For example, Chow et al. [14] leverage Lyapunov stability principles to obtain improved empirical results. Other methods focus on safety as a challenge relevant to transfer learning, where safe behavior can be extrapolated and expanded from simpler tasks [15]. Methods in the space of RL provide idealistic safety guarantees that translate into improved empirical safety properties. However, any guarantees (probabilistic or robust) or safety certificates in this space are elusive and remain an open challenge.

Guarantees in RL literature are difficult to obtain since that literature eschews SME, or direct intuition into a specific application. Some RL research obtains guarantees by leveraging strong SME in the form of known safe backup controllers [16], [17]. Generally, when RL neglects considerations to SME, it becomes applicable to a much wider body of relevant decision and control problems [18] that lack permeability to our intuition and expertise. Conversely, control literature is ubiquitous in revealing how such expertise can be leveraged to yield strong and *specific* performance and safety even in adaptive and learning contexts. As previously discussed, SME in controls LbC methods often takes the form of model knowledge [5], [6], [9], [10], [11] and preexisting data of safe trajectories [7], [19].

The problem with these SME assumptions is that they can very easily become optimistic. Given the overarching assumption of preexisting data of safe trajectories, we have to ask "How trustworthy is our data?" This should always be called into question, especially when safety is of the utmost importance. Many LbC methods do consider noise-corrupted data [19], but what if deeper, malicious pathology infiltrates the data generation process? The process of generating the data could be flawed in many ways, the relevance of each to existing methods varies but is persistent. An example could be sampling data locally where relevant dynamics can be effectively linearized when the system experiences highly nonlinear behavior outside of that region. Without exploiting and trusting our SME, we cannot guarantee things like this will not happen especially in safety-critical settings. By applying a resultant controller to the underlying system, it can encounter out-of-distribution (OOD) experience and adversarial attacks that a majority of existing LbC methods simply cannot accommodate. These few LbC algorithms that do make consideration to OOD experience do so using hyperparameters that are not trivial to select and validate [19] and often assume the structure of the underlying dynamics [20]. These same fundamental quandaries also apply when assuming model knowledge.

In this article, we address these key open questions about SME in control-theoretic LbC. Critically, we ask "*What is the least amount of SME we may need to obtain safe control results?*" Such questions remain relatively unexplored in control literature, despite their relevance. Our methods for addressing these questions are actually quite simple and rely on a combination of concepts in stochastic MPC and distributionally robust optimization (DRO). We make this technical augmentation along with several basic assumptions about the problem formulation that allows us to translate probabilistic safety guarantees in the absence of conventionally strong dependence on SME.

A. Background on DRO and LbC

This article primarily leverages concepts from DRO to obtain safety certificates. In recent practice, DRO has been gaining traction as a set of methods that provide significant value to the study and solution of the LbC problem. DRO is a field of inquiry that seeks to guarantee robust solutions to optimization programs when the distributions of relevant random variables are estimated via sampling. This uncertainty can involve the objective or the constraints of the optimization program. Uncertainty in both cases can pose significant challenges if unaccounted for, leading to suboptimal and potentially unsafe performance [21]. Given that past work in the LbC space frequently considers chance constraints [5], [19], [22], incorporating a true DRO approach possesses the potential to improve our capabilities of guaranteeing safety during learning. These methods have been recently explored to address challenges of safety and performance imposed by uncertainty. For instance, Van Parys et al. [23] addressed the distributional uncertainty of a random exogenous disturbance process with a moment-based framework. Paulson et al. [24] also applied polynomial chaos expansions to characterize distributional parametric uncertainty in a nonlinear model-predictive control application.

Within the toolbox provided by DRO, Wasserstein ambiguity sets are a foremost asset. The Wasserstein metric (or "earth mover's distance") is a symmetric distance measure in the space of probability distributions. Wasserstein ambiguity sets account for distributional uncertainty in a random variable, frequently one approximated in a data-driven application. They accomplish this feat with out-of-sample performance guarantees by replacing the data-driven distribution of the random variable with the worst-case realization within a Wasserstein ball centered on the empirical distribution [25], [26]. Expressions exist that map the quality of the empirical distribution with Wasserstein ball radii such that desired robustness characteristics are achieved without significant sacrifices to the performance of the solution [27]. Within the control context, however, the Wasserstein distance metric has only recently begun emerging as a valuable and widespread tool. Yang [28] explored the application of Wasserstein ambiguity sets for distributionally robust control subject to disturbance processes. Similar methods have made their way to research on model-based and model-free RL as well [20], [29], [30]. DRO has also been applied to Markov decision processes (MDPs) in a general sense, with good results [31], [32], [33], [34]. Scalability is still an open challenge in that space. Overall, while Wasserstein ambiguity sets are seeing increased application in control research, many of their true capabilities have yet to be fully exploited.

B. Statement of Contributions

This article seeks to address key shortcomings in these areas of literature. Among those previously discussed, foremost is the lack of general methods that possess robustness when conducting *tabula-rasa* LbC, or those requiring significant assumptions on the availability of prior data of safe control trajectories.

We present a novel and simple model-based LbC scheme based on MPC which provides strong probabilistic out-of-sample guarantees on safety. We validate our method using experiments that emulate *tabula-rasa* as closely as possible given our assumptions, but our algorithm is widely applicable to adaptive control scenarios especially when underlying dynamics may be poorly structured or difficult to characterize. By developing Wasserstein ambiguity sets relating to empirical distributions of modeling error, we can conduct MPC with an imperfect learned snapshot model while maintaining confidence in our ability to satisfy nominal constraints. The Wasserstein ambiguity sets allow us to optimize with respect to constraint boundaries that are shifted into the safe region. As our empirical distributions improve with more data, the offset variables tighten toward the nominal boundary in a provably safe way. Our approach yields probabilistic safety guarantees. Critically, in this article, we present this LbC scheme along with: 1) an explicit and fundamental PoE scheme and 2) highly limited SME assumptions. While many LbC methods are amenable to PoE schemes [4], the question of PoE is in some cases neglected despite its relevance. We actually show our explicit PoE scheme is fundamental to illustrating the applicability of our method. Our contributions combine to allow us to translate safety guarantees with highly limited model knowledge and data.

The overarching objective of this article is not to present the most high-performing LbC architecture, but rather to explore what kind of performance we can obtain when limiting our SME assumptions more so than existing work in controls literature. Many control-theoretic methods provide stronger robust (i.e., safety w.p. 1) guarantees under much more restrictive assumptions. In our case, we label our method as "trustworthy" insofar as it relies on highly limited SMEs. Given the elusiveness of safety guarantees in RL literature, a probabilistic result within our context is powerful.

We validate our approach with two case studies focusing on safety-critical applications. In the first, we learn a policy that safely charges a lithium-ion battery using a nonlinear equivalent circuit model (ECM). Battery fast charging presents a strong challenge for LbC methods, given that the optimal policy is a boundary solution that rides constraints until the terminal conditions are met. We also conducted a case study on safe autonomous driving using a nonlinear bicycle model of vehicle dynamics. We demonstrate that our algorithm provides a provably safe method for the vehicle to avoid obstacles while learning its dynamics from scratch.

We provide an open-source GitHub repository [35] for our case studies.

II. Distributionally Robust Optimization

The core of our proposed algorithmic architecture relies heavily on DRO techniques. In this section, we outline fundamental ideas which establish the foundation of our algorithm.

A. Chance-Constrained Programming

A chance constraint is a constraint within an optimization program that is only satisfied with some probability. This is typically a necessary concession when the constraint is affected by a random variable $\mathbf{R}$

$$\mathbb{P}[h(x_k, u_k, \mathbf{R}) \leq 0] \geq 1 - \eta. \tag{1}$$

Here, the constraint function $h(x_k, u_k, \mathbf{R})$ outputs an m-dimensional vector. In this case, the distribution $\mathbb{P}$ relates to random variable $\mathbf{R}$ with support ξ. Here, $0 \leq \eta < 1$ is the specified risk metric or our allowed probability to violate the constraint. If $\eta = 0$, we say we have a robust optimization program that must not yield *any* probability of constraint violation. In practice, especially when approximating $\mathbb{P}$ from sampling, we admit some small probability of constraint violation leading to a value of $\eta > 0$. This is frequently necessary because it allows our probabilistically robust solution to balance conservatism with performance.

Upon utilizing an empirical approximation of $\mathbb{P}$ derived from sampling (usually denoted $\hat{\mathbb{P}}$), we admit some distributional uncertainty that can arise from only having access to a finite group of samples. The law of large numbers states that for any number of samples $\ell \to \infty$, $\hat{\mathbb{P}} \to \mathbb{P}^*$. The discrepancy from this limited sampling creates distributional uncertainty, which can affect the quality of the solution if our approximation $\hat{\mathbb{P}}$ is inaccurate [21]. Throughout the remainder of this section, we discuss the application of DRO techniques to address this distributional uncertainty.

B. Wasserstein Ambiguity Sets

The Wasserstein metric is defined as follows.

Definition 1: Given two marginal probability distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ lying within the set of feasible distributions $\mathcal{P}(\xi)$, the Wasserstein distance between them is defined by

$$\mathcal{W}(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\Pi} \left\{ \int_{\xi^2} \|\mathbf{R}_1 - \mathbf{R}_2\|_a \Pi(d\mathbf{R}_1, d\mathbf{R}_2) \right\} \tag{2}$$

where Π is a joint distribution of the random variables $\mathbf{R_1}$ and $\mathbf{R_2}$, and a denotes any norm in $\mathbb{R}^n$.

The Wasserstein metric is colloquially referred to as the "earth-movers distance." This name is rooted in the interpretation of the Wasserstein metric as the minimum cost of redistributing mass from one distribution to another via nonuniform perturbation [28]. To show why the Wasserstein distance is a valuable tool we can leverage to robustify a data-driven optimization program, we first reference the chance constraint (1), which depends on an empirical distribution $\hat{\mathbb{P}}$. Rather than solving the optimization program with respect to an imperfect snapshot of $\mathbb{P}^*$ defined by $\hat{\mathbb{P}}$, we can optimize over any probability distribution within some ambiguity set centered around our estimate $\hat{\mathbb{P}}$. The Wasserstein distance provides a formal method to define such an ambiguity set, namely we can optimize against the worst-case realization of $\mathbf{R}$ sourced from a set of probability distributions within the specified Wasserstein radius of our empirical estimate. We define "worst-case" as the realization that yields the lowest probability of satisfying the chance constraint. This formulation can be described mathematically with the following relation:

$$\inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}[h(x_k, u_k, \mathbf{R}) \leq 0] \geq 1-\eta \tag{3}$$

where

$$\mathbb{B}_\epsilon := \left\{\mathbb{P} \in \mathcal{P}(\xi) \mid \mathcal{W}\left(\mathbb{P}, \hat{\mathbb{P}}\right) \leq \epsilon\right\} \tag{4}$$

is the ambiguity set defined for a Wasserstein ball radius ϵ. Of note is the fact that (3) guarantees probabilistic feasibility for any probability distribution within the ambiguity set when reformulated correctly. No assumptions must be leveled on the true distribution $\mathbb{P}^*$ for these guarantees to translate under a proper reformulation.

Reformulation is necessary because the exact constraint shown in (3) poses an infinite-dimensional nonconvex problem. Ongoing research has pursued tractable reformulations of this constraint which facilitate its real-time solution.

This article adopts a reformulation of (3) detailed in [36]. This reformulation accommodates vector constraint functions and requires that the function $g(x_k, u_k, \mathbf{R})$ is linear in $\mathbf{R}$ and entails a scalar convex optimization program to derive. Our algorithm is designed to exploit the linear dependence on R such that this assumption does not affect the applicability of our approach. Importantly, the result is a conservative *convexity-preserving* approximation of (3). For an m-dimensional constraint function, the exact form of the ambiguity set is $\mathcal{V} = \text{conv}(\{r^{(1)}, \ldots, r^{(2^m)}\})$, where the vector r is sourced from the optimization component of the overall procedure. The set of constraints we find to replace the infinite-dimensional DRO chance constraint are

$$h(x_k, u_k) + r^{(j)} \leq 0 \quad \forall j = 1, \ldots, 2^m. \tag{5}$$

For a complete and elegant discussion of this reformulation, we highly recommend the reader reference work in [36], specifically pages 5–7 of their article. This reformulation requires some additional information, including a tractable representation of an appropriate Wasserstein ball radius.

Finally, several expressions exist for the Wasserstein ball radius ϵ which are probabilistically guaranteed to contain the true distribution with allowed probability β. We adopt the following formulation of ϵ from [27]:

$$\epsilon(\ell) = C\sqrt{\frac{2}{\ell}\log\left(\frac{1}{1-\beta}\right)} \tag{6}$$

where ℓ is the number of data points, β is the probability the Wasserstein ball contains the true distribution, and C relates to the diameter of the support of the distribution and is obtained by solving the following scalar optimization program:

$$C \approx 2 \inf_{\alpha>0}\left\{\frac{1}{2\alpha}\left(1+\ln\left(\frac{1}{\ell}\sum_{k=1}^{N} e^{\alpha\|\mathbf{R}^k-\hat{\mu}\|_1^2}\right)\right)\right\}^{\frac{1}{2}} \tag{7}$$

where the right side bounds the value of C, $\mathbf{R}^k$ is a sample of the random variable that comprises our empirical distribution, and $\bar{\mu}$ is the sample mean of the distribution.

III. Equivalent Chance-Constraint Reformulation

This article builds upon the equivalent reformulation of (3) from [36]. This reformulation leverages findings from recent work by [25]. The statement of the specific reformulation in [36] indicates a requirement that the constraint function $g(x, \mathbf{R})$ is linear in x and R, respectively.

Notably, we identify a simple extension of the reformulation in [36] that allows its application to our nonlinear MPC formulation via relaxing the requirement the constraint function be linear in the decision variable x.

A. Restatement of the Reformulation From [36]

The reformulation from [36] is stated to require the constraint function $g(x, \mathbf{R})$ to be linear in x and $\mathbf{R}$, respectively. In the next subsection, we extend the reformulation to include some broader cases of constraint functions

$$g(x, \mathbf{R}) = g_x(x) + g_R(\mathbf{R}) \tag{8}$$

where functions g_x and g_R can be nonlinear in their respective arguments. In this section, we restate the work from [36] as a reference for our extension included in Section III-B.

Data samples $\{R^{(1)}, R^{(2)}, \ldots, R^{(\ell)}\}$ corresponding to random variable $\mathbf{R} \in \mathbb{R}^{\mathbf{m}}$ are drawn from the true distribution $\mathbb{P}^*$. These finite samples comprise our empirical distribution $\hat{\mathbb{P}}$. The finiteness of our empirical distribution indicates that it will not perfectly match the behavior of the true distribution $\mathbb{P}^*$. This is especially true in cases with limited samples, which are relevant to the challenging case studies this article explores.

Normalizing the data lends simplicity to the derivation

$$\vartheta^{(i)} = \Sigma^{-\frac{1}{2}}\left(R^{(i)} - \mu\right) \tag{9}$$

where Σ is the sample variance of the data and μ is the sample mean. This standardization transforms the data samples such that its new mean is 0, and its new variance is $I_{m\times m}$. The support of this normalized distribution is

$$\Theta = \left\{\vartheta \in \mathbb{R}^m \mid -\sigma_{\max}\mathbf{1}_m \leq \vartheta \leq \sigma_{\max}\mathbf{1}_m\right\} \tag{10}$$

since we have centered the normalized variable ϑ. Note that $\mathbf{1_m}$ is a column vector of ones. Let $\mathbb{Q}^*$ and $\hat{\mathbb{Q}}$ represent the true and empirical distributions of the normalized data ϑ. We construct the ambiguity set $\hat{\mathcal{Q}}$ using the "Wasserstein ball" given by (4), allowing us to transform the distributionally robust chance constraint (DRCC) in (3) to

$$\sup_{\mathbb{Q}\in\hat{\mathcal{Q}}} \mathbb{Q}[\vartheta \notin \mathcal{V}] \leq \eta \tag{11}$$

which says the worst-case probability that normalized random variable ϑ is outside set $\mathcal{V}$ is less than η, where the supremum is taken over all distributions $\mathcal{Q}$ in ambiguity set $\hat{\mathcal{Q}}$. We wish to obtain the least conservative (i.e., tightest) set $\mathcal{V} \subseteq \mathbb{R}^m$

to define the desired Wasserstein uncertainty set $\mathcal{A} = \{a \in \mathbb{R}^m \mid a = \Sigma^{(1/2)} v + \mu, v \in \mathcal{V}\}$ such that

$$g(x_k, u_k, \mathbf{R}) \le 0 \quad \forall \mathbf{R} \in \mathcal{A}. \tag{12}$$

We restrict the overall shape of the set $\mathcal{V}$ to be a hypercube, which enables computational tractability

$$\mathcal{V}(\sigma) = \left\{\vartheta \in \mathbb{R}^m \mid -\sigma 1_m < \vartheta < \sigma 1_m\right\}. \tag{13}$$

Now, to compute this ambiguity set without introducing unnecessary conservatism, we need to find the minimum value of the hypercube side length $\sigma \in \mathbb{R}$. The following optimization program details this problem:

$$\min_{0 \le \sigma \le \hat{\sigma}_{\max}} \quad \sigma \tag{14}$$

$$\text{s.t.: } \sup_{\mathbb{Q} \in \hat{\mathcal{Q}}} \mathbb{Q}\left[\tilde{\vartheta} \notin \mathcal{V}(\sigma)\right] \le \eta. \tag{15}$$

Here, we select $\hat{\sigma}_{\max}$ using a priori information about the specific problem context.

The derivation in [36] provides a worst-case probability formulation, summarized by the following lemma.

Lemma 1: [36, Lemma 2]

$$\sup_{\mathbb{Q} \in \hat{\mathcal{Q}}} \mathbb{Q}\left[\tilde{\vartheta} \notin \mathcal{V}(\sigma)\right] = \inf_{\lambda \ge 0} \left\{ \lambda \epsilon(\ell) + \frac{1}{\ell} \sum_{j=1}^{\ell} \left(1 - \lambda\left(\sigma - \left\|\vartheta^{(j)}\right\|_\infty\right)^+\right)^+ \right\} \tag{16}$$

where $(x)^+ = \max(x, 0)$.

We defer to [36] for the proof of this finding. Their result entails that (16) can be reformulated as

$$\min_{0 \le \lambda, 0 \le \sigma \le \hat{\sigma}_{\max}} \quad \sigma \quad \text{s.t.: } h(\sigma, \lambda) \le \eta \le \sigma_{\max} \tag{17}$$

where

$$h(\sigma, \lambda) = \lambda \epsilon(\ell) + \frac{1}{\ell} \sum_{j=1}^{\ell} \left(1 - \lambda\left(\sigma - \left\|\vartheta^{(j)}\right\|_\infty\right)^+\right)^+. \tag{18}$$

The result of this optimization program is the value of σ, which is used to reformulate the chance constraints via convex approximation. For a convex approximation of the constraint function in (3), the hypercube $\mathcal{V}(\sigma)$ becomes the convex hull of its vertices. If, for example, $m = 1$ (i.e., the random variable is 1-D), then $\mathcal{V}(\sigma) = (-\sigma, \sigma)$—an open interval. The offset $r^{(j)}$ is calculated from

$$r^{(1)} = \Sigma^{\frac{1}{2}} \mathbf{1}_m \sigma + \mu \tag{19}$$

$$r^{(2)} = \Sigma^{\frac{1}{2}} \mathbf{1}_m (-\sigma) + \mu. \tag{20}$$

In the 2-D case, this yields the ambiguity set $\mathcal{A} = \text{conv}(\{\pm\sigma, \pm\sigma\})$, where $\text{conv}(\{\cdots\})$ represents the convex hull of points $\{\cdots\}$. For an m-dimensional constraint function, the exact form of the ambiguity set is $\mathcal{V} = \text{conv}(\{r^{(1)}, \ldots, r^{(2^m)}\})$. In each case, the ambiguity set is a hypercube, and the change of signs is the method by which we enumerate across that hypercube's vertices with the following constraints:

$$g(x) + r^{(j)} \le 0 \quad \forall j = 1, \ldots, 2^m. \tag{21}$$

Algorithm 1 details the method used to compute the offset σ.

Algorithm 1 Computation of σ

```
Initialize $\underline{\sigma} = 0, \bar{\sigma} = \sigma_{max}$
while $\bar{\sigma} - \underline{\sigma} >$ tolerance do
    $\sigma = \frac{\bar{\sigma} + \underline{\sigma}}{2}$
    $[\lambda, h^*(\sigma, \lambda)]$ = minimize($\sigma$, $\lambda_{lb}$, $\lambda_{ub}$, $\epsilon$, $\theta$)
    if $h^*(\sigma, \lambda) > \eta$ then
        $\underline{\sigma} = \sigma$
    else
        $\bar{\sigma} = \sigma$
    end if
end while
$\sigma = \bar{\sigma}$
```

B. Extending the Reformulation

Esfahani and Kuhn [25] utilized the findings in presenting their convex reformulation. Critically, we identify that the fundamental theory presented by [25] allows applying the identical reformulation to cases where the constraint function takes the form

$$g(x, \mathbf{R}) = g_x(x) + g_R(\mathbf{R}) \tag{22}$$

where g_x and g_R may be nonlinear functions. Critically, there must not be any interdependence between x and $\mathbf{R}$.

This article presents a modified lemma for the applicability of the previously stated reformulation first presented by [36].

Lemma 2: If the function g satisfies

$$g(x, \mathbf{R}) = g_x(x) + g_R(\mathbf{R}) \tag{23}$$

then constraints of the following form

$$\inf_{\mathbb{P} \in \mathbb{B}_\epsilon} \mathbb{P}\left[g(x, \mathbf{R}) \le 0\right] \ge 1 - \eta \tag{24}$$

can be reformulated into the convex approximation

$$g_x(x) + r^{(j)} \le 0 \quad \forall j = 1, \ldots, 2^m \tag{25}$$

using the relations in (16) and (17), where $r = \Sigma^{(1/2)} \mathbf{1}_m \sigma + \mu$.

Proof: We start by defining auxiliary variables in the constraint function. Consider that, without loss of generality, nonlinear functions of $\mathbf{R}$ can themselves be considered the random variable in question

$$\tilde{\mathbf{R}} = g_R(\mathbf{R}) \tag{26}$$

where $\tilde{\mathbf{R}}$ is the new model of the stochasticity. This gives

$$g(x, \mathbf{R}) = g_x(x) + \tilde{\mathbf{R}}. \tag{27}$$

Now, we create a dummy auxiliary decision variable $\tilde{x}$ in the same manner

$$\tilde{g}(\tilde{x}, \tilde{\mathbf{R}}) = \tilde{x} + \tilde{\mathbf{R}} \tag{28}$$

forming a function $\tilde{g}$ which is trivially linear in $\tilde{x}$ and $\tilde{\mathbf{R}}$, where

$$\tilde{x} = g_x(x). \tag{29}$$

This equality constraint (29) now shows up in the overall optimization program. However, the DRCC reformulation only poses conditions on the constraint function in question [namely $\tilde{g}(\tilde{x}, \tilde{\mathbf{R}})$]. We have transformed the DRCC into

$$\inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}\big[\tilde{g}\big(\tilde{x}, \tilde{\mathbf{R}}\big) \leq 0\big] \geq 1-\eta \tag{30}$$

which is now linear in $\tilde{x}$ and $\tilde{\mathbf{R}}$. Following procedure from [25], we suppress dependence on x (or $\tilde{x}$) for simplicity, leading to $\ell(\tilde{\mathbf{R}}) = \tilde{g}(\tilde{x}, \tilde{\mathbf{R}})$ [25], [36]:

$$\inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}\big[\ell\big(\tilde{\mathbf{R}}\big) \leq 0\big] \geq 1-\eta. \tag{31}$$

The remainder of the proof is identical to the Appendix in [36], leading to the convex approximation

$$g_x(x) + r^{(j)} \leq 0 \quad \forall j = 1, \dots, 2^m. \tag{32}$$

Beyond exploiting the linear presence of $\tilde{x}$ in the constraint function, suppressing dependence on decision variables is possible and helpful for the following reasons. The overall process of solving an optimization program with a DRCC is characterized by a two-stage stochastic optimization problem. Here, (31) is the first stage problem that we solve using the equivalent reformulation. Esfahani and Kuhn [25] showed in Section V-C of their article that, without loss of generality, the solution in the second stage (i.e., the overall optimization program) is unaffected by suppressing dependence of ℓ on decision variables in the first stage. In addition, the decision-independent loss function $\ell(\tilde{\mathbf{R}})$ can trivially be expressed as a pointwise maximum of elementary measurable functions, as required by Section IV.

In practice, the dummy decision variable $\tilde{x}$ will not come into play during any stage of the solution. After solving the first-stage problem, we can reverse the substitution in the remaining optimization to avoid an equality constraint with poor computational tractability.

Remark 1: The linear separability of x and R poses a DRO program that can be thought of as solving

$$\min_{z\in\mathbb{R}} \; -z \tag{33a}$$

$$\text{s.t.:} \inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}[z - R \leq 0] \geq 1-\eta \tag{33b}$$

for offset z from the second-stage nominal constraint boundary.

We have shown a simple extension of the DRO reformulation from [36] that allows us to apply the method to nonlinear optimization programs. In Section IV of this article, we describe our nonlinear MPC formulation and the context within which the guarantee from the DRCC is translated to LbC.

IV. Distributionally Robust Model-Based LbC

Fig. 1 shows a block diagram of our proposed control architecture, detailed within this section.

A. Model Predictive Control Formulation

We apply Wasserstein ambiguity sets to robustify a learning model predictive controller, based on the following optimization program formulation. Given true plant dynamics

$$x_{t+1} = f(x_t, u_t, W_t) \tag{34}$$

$$y_t = g(x_t, u_t, V_t) \tag{35}$$

where t is the current timestep, W_t is the state noise, V_t is the output measurement noise, x_t is the state variable, and y_t is the output variable. We assume access to full state and output measurements, subject to the measurement noises W_t and V_t. The capital letters represent random variables. Before considering modifications for distributional robustness to uncertainty (which also accommodate exogenous inputs), we seek to solve the following predictive control problem:

$$\min_{u_{t:t+N-1}} \sum_{k=t}^{t+N} J_k\big(\hat{x}_k, \hat{y}_k, u_k\big) \tag{36a}$$

$$\text{s.t.:} \tag{36b}$$

$$\hat{x}_{k+1} = \hat{f}\big(\hat{x}_k, u_k, \theta_f\big) \tag{36c}$$

$$\hat{y}_k = \hat{g}\big(\hat{x}_k, u_k, \theta_g\big) \tag{36d}$$

$$\hat{y}_k \leq 0 \tag{36e}$$

$$\hat{x}_t = x_t \tag{36f}$$

where x_t is the known (measured) initial state at the current timestep t. The *"hat"* symbol indicates a predicted variable, and the learned models themselves are given by

$$\hat{x}_{t+1} = \hat{f}\big(x_t, u_t, \theta_f\big) \tag{37}$$

$$\hat{y}_{t+1} = \hat{g}\big(x_t, u_t, \theta_g\big). \tag{38}$$

At a high level, these can be thought of as two separate models. However, when learning a black-box representation of the system, that single model can be trained to predict both sets of values $\hat{x}_{t+1}$ and $\hat{y}_t$. The parameters θ_f and θ_g are learned from historical data through model identification.

B. Model Identification

The models are used to predict state transition dynamics and constraint function outputs. We assume that the true model parameters θ_f^* and θ_g^* are inaccessible to the controller. Several methods can be selected to learn the parameters online and can depend on what type of learning model architecture is selected. In this article, we utilize nonlinear least-squares with neural network models for both the state transition dynamics and constraint functions

$$\hat{f}\big(x_t, u_t, \theta_f\big) \leftarrow x_{t+1} \tag{39}$$

$$\hat{g}\big(x_t, u_t, \theta_g\big) \leftarrow y_t \tag{40}$$

where x_{k+1} and y_k are assumed to be measurable from the real system at the current timestep. When conducting MPC, the initial x_k is obtained by assuming full state observability throughout the LbC problem. From this point forward, we denote $\theta_{g;t}$ as the parameterization of the learned model of g at timestep t in the overall learning process.

C. Modeling Error Characterization

We characterize modeling error through comprehensive modeling residuals across varying prediction depths.

For example, consider a scalar system $x \in \mathbb{R}$, $y \in \mathbb{R}$ within three steps of model predictive control $N = 2$ with quadratic,

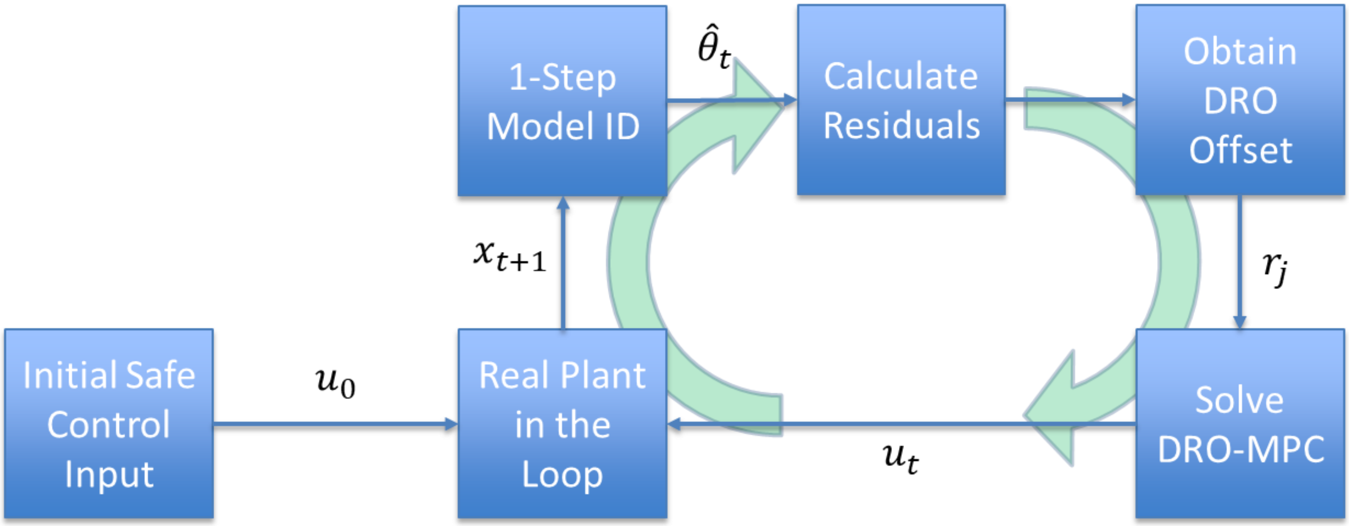

Fig. 1. Diagram of safe Wasserstein-constrained MPC. In the most restrictive case, after initializing the controller, it immediately begins interacting with its environment. At every timestep, it observes an MDP state transition tuple, calculates model residuals, uses the residuals to calculate the DRO offset $r^{(j)}(k)$, and then solves a new MPC program at the next state. This application case serves as a purposefully extreme challenge to the robustness and behavior of our algorithm at what would otherwise be unreasonable levels of uncertainty and risk. Later in our article, we demonstrate that even under such extreme conditions, we manage to safely learn control policies for a host of nonlinear stochastic control problems. We do note, however, that our algorithm is much more widely applicable when prior data and SME is available.

time-invariant objective function (state penalty $q = 1$, effort penalty $r = 1$, and terminal state penalty $p = 1$)

$$\min_{u_t, u_{t+1}, u_{t+2}} \quad x_t^2 + \hat{x}_{t+1}^2 + u_t^2 + u_{t+1}^2 + \hat{x}_{t+2}^2 \tag{41a}$$
$$\text{s.t.:} \tag{41b}$$
$$\hat{x}_t = x_t \tag{41c}$$
$$\hat{x}_{t+1} = \hat{f}(x_t, u_t, \theta_f) \tag{41d}$$
$$\hat{x}_{t+2} = \hat{f}(\hat{x}_{t+1}, u_{t+1}, \theta_f) \tag{41e}$$
$$\hat{x}_{t+3} = \hat{f}(\hat{x}_{t+2}, u_{t+2}, \theta_f) \tag{41f}$$
$$\hat{y}_t = \hat{g}(x_t, u_t, \theta_g) \tag{41g}$$
$$\hat{y}_{t+1} = \hat{g}(\hat{x}_{t+1}, u_{t+1}, \theta_g) \tag{41h}$$
$$\hat{y}_{t+2} = \hat{g}(\hat{x}_{t+2}, u_{t+2}, \theta_g) \tag{41i}$$
$$\hat{y}_t \leq 0 \tag{41j}$$
$$\hat{y}_{t+1} \leq 0 \tag{41k}$$
$$\hat{y}_{t+2} \leq 0. \tag{41l}$$

Suppose we find a sequence u_t^*, u_{t+1}^*, u_{t+2}^* from solving three sequential model predictive control problems with the true plant in the loop. Since we are using learned models to solve these predictive control problems, these inputs are likely not actually optimal for the system, and with added PoE, they include exploratory aspects. In each case, we apply the first control input to the system to obtain x_{t+1}^*, x_{t+2}^*, x_{t+2}^*. We can quantify the prediction error of the learned constraint function in the following manner:

$$R_1^{(t)} = g(x_t, u_t^*) - \hat{g}(x_t, u_t^*, \theta_g) \tag{42a}$$
$$R_1^{(t+1)} = g(x_{t+1}^*, u_{t+1}^*) - \hat{g}(\hat{x}_{t+1}, u_{t+1}^*, \theta_g) \tag{42b}$$
$$R_1^{(t+2)} = g(x_{t+2}^*, u_{t+2}^*) - \hat{g}(\hat{x}_{t+2}, u_{t+2}^*, \theta_g). \tag{42c}$$

These are one-step residuals, as denoted by the subscript R_1, since $\hat{x}_{t+1} = f(x_t, u_t^*)$ and $\hat{x}_{t+2} = f(x_{t+1}^*, u_{t+1}^*)$. In these equations, the function g represents our observations from the real system (simple data), and the function $\hat{g}$ represents the predictions of our learned constraint model. We take the absolute value since these residuals will be introduced as variables that add conservatism relative to the existing constraint boundary. Since we conduct predictive control, we also want to quantify modeling errors after 2, 3, or more steps of prediction into the future using learned models, as errors can accumulate and become worse with successive prediction steps. This happens in the following way:

$$R_1^{(t)} = \left|g(x_t, u_t^*) - \hat{g}(x_t, u_t^*, \theta_g)\right| \tag{43a}$$
$$R_2^{(t)} = \left|g(x_{t+1}^*, u_{t+1}^*) - \hat{g}(\hat{f}(x_t, u_t^*, \theta_f), u_{t+1}^*, \theta_g)\right| \tag{43b}$$
$$R_3^{(t)} = \left|g(x_{t+2}^*, u_{t+2}^*)\right. \tag{43c}$$
$$\left. - \hat{g}(\hat{f}(\hat{f}(x_t, u_t^*, \theta_f), u_{t+1}^*, \theta_f), u_{t+2}^*, \theta_g)\right|. \tag{43d}$$

As is shown here, modeling error accumulates from the learned representation of both the constraint function $\hat{g}$ and the learned dynamics function $\hat{f}$.

Remark 2: We choose to take the absolute value of residuals. This decision is not necessary but makes intuitive sense given the application. Since we intend to modify the nominal constraint boundary, signals of modeling errors that show underestimation could lead to an offset that potentially moves the constraint into the unsafe region. We seek to avoid this and only create offsets that reduce the size of the feasible region.

The model identification process utilizes the one-step residuals to minimize the mean-square prediction error (mse) of the prediction of the state transition compared to past observations. The multistep residuals are utilized by the DRO framework to adjust conservatism deeper into the future based on cumulative modeling error.

By representing modeling error this way, we lump all relevant sources of modeling error into an additive term.

As previously discussed, the absolute value is taken as a precautionary measure. Omitting that transformation provides the following simple expression:

$$g\big(x^*_{t+2}, u^*_{t+2}\big) = \hat{g}\big(\hat{f}\big(\hat{f}\big(x_t, u^*_t, \theta_g\big), u^*_{t+1}, \theta_g\big), u^*_{t+2}, \theta_g\big) + R_3^{(t)}. \tag{44}$$

By treating the residuals as random variables drawn from a true distribution $\mathbb{P}$, the constraints will by definition be additive in the random variable/modeling error.

D. Safety and Robustness Using Wasserstein Ambiguity Sets

Now that we have outlined the distributionally robust chance-constrained approach using the Wasserstein ambiguity set, we can describe how it fits within our robust control framework.

The residuals defined in Section IV-C entail a representation of the modeling error. This is only true because the constraint functions are evaluated using predicted states from the learned dynamical model, whose true representation is unknown. By considering process error/residuals as an additive noise term, we can maximize the utility of the DRO reformulation in [36] which requires this linear structure in the constraint

$$g\big(x_k, u_k, \theta_{g;t}\big) + \mathbf{R_1} \leq 0. \tag{45}$$

As previously discussed and shown in (44), by design, this linear structure will always occur. These residuals are random variables characterized by empirical distributions based on our observations. Now, we have bolded the variable $\mathbf{R_1}$ to indicate it is a random variable, whereas the previous value $R_1^{(t)}$ was a realization of this random variable at time t.

To accommodate distributional uncertainty in our estimate of $\hat{\mathbb{P}}$, we transform the constraint (45) for each of $1 \rightarrow N+1$ step residuals into a joint DRCC via Wasserstein ambiguity set as follows:

$$\inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}\begin{bmatrix} \hat{g}\big(\hat{x}_k, u_k, \theta_{g;t}\big) + \mathbf{R_1} \leq \mathbf{0} \\ \hat{g}\big(\hat{x}_{k+1}, u_{k+1}, \theta_{g;t}\big) + \mathbf{R_2} \leq \mathbf{0} \\ \vdots \\ \hat{g}\big(\hat{x}_{k+N}, u_{k+N}, \theta_{g;t}\big) + \mathbf{R_{N+1}} \leq \mathbf{0} \end{bmatrix} \geq 1 - \eta. \tag{46}$$

The reformulation we adopt from [36] presents a simple method to accommodate the constraint without inverting the CDF. If we operate under the assumption that the residuals for $i = 1, \ldots, N$ steps are uncorrelated, then we can decompose this joint chance constraint into a set of individual chance constraints. This decomposition could be useful if the optimization algorithm we select to solve the MPC problem scales unfavorably with the dimension of the constraints. Algorithm 1 provides an overview of the real-time implementation of our approach. As previously stated, the process for computing r entails a simple scalar convex optimization program.

Remark 3: The reformulation from [36] adds cardinality of constraints that scale with order 2^m. However, our formulation of modeling error as an additive residual allows the number of constraints to remain constant. We detail this property in the Appendix of this article. The simple answer is that, by taking the absolute values of the residuals, the random variable that represents modeling error is strictly nonnegative. This means that a negative realization is impossible to encounter and need not be accommodated. By keeping the cardinality of constraints constant, the computational scalability of our approach is preserved for higher-dimensional control problems.

Algorithm 2 Wasserstein Robust Learned MPC

```
Require: State space §, Action space $\mathcal{U}$
  for $t$ in range $t_{max}$ do
    if $t = 1$ then
      $u_t$ = known safe input, $N = 1$
    else
      Update the dynamical system model and constraint functions $\theta_{t-1} \rightarrow \theta_t$
      Receding horizon increment rule (i.e. $N = min\{N_{targ}, round(\frac{t}{N_{targ}}) + 1\}$)
      Obtain Wasserstein ambiguity set offset $r$:
      $u_t \leftarrow$ Solve MPC optimization program (48a)-(48i)
    end if
    $x_{t+1} = f(x_t, u_t, W_t)$ (Truth plant)
    $y_t = g(x_t, u_t, V_t)$ (Truth plant)
  end for
```

At each time step, we compute model residuals with our most recent estimate $\theta_{g;t}$ using predicted state transitions from our entire cumulative experience, compile a unique empirical distribution $\hat{\mathbb{P}}$ corresponding to each individual chance constraint, and compute the value of r in (5) to reformulate the DRCCs. We can begin the overall process with a small control horizon N and gradually increase N as we accumulate more and more data from experience. The residuals we compute are for horizon lengths of 1 to N-steps, meaning the elements of $\mathbf{R}$ correspond to each of $i = 1, \ldots, N$ step residuals. Then, we assemble a joint chance constraint where the elements of the column vector of the random variable are the $1 \rightarrow N$ step residuals. In [36], the authors pursued a DRO reformulation that utilizes a polytopic representation of the uncertainty set. Our formulation preserves scalability by isolating dependence on the random variable in the constraint. Our appendix shows the logic that allows the cardinality of constraints to remain constant.

Finally, when we conduct MPC, we replace the nominal constraints with their distributionally robust counterparts

$$\min_{u\in\mathcal{U}} \sum_{k=t}^{t+N} J_k\big(\hat{x}_k, u_k\big) \tag{47a}$$

$$\text{s.t.: } \hat{x}_{k+1} = \hat{f}\big(\hat{x}_k, u_k, \theta_{g;t}\big) \tag{47b}$$

$$\begin{bmatrix} \hat{g}\big(\hat{x}_k, u_k, \theta_{g;t}\big) \\ \hat{g}\big(x_{k+1}, u_{k+1}, \theta_{g;t}\big) \\ \vdots \\ \hat{g}\big(\hat{x}_{k+N}, u_{k+N}, \theta_{g;t}\big) \end{bmatrix} + r^{(j)} \leq 0 \tag{47c}$$

$$\hat{x}_0 = x_t. \tag{47d}$$

Algorithm 2 describes the implementation of our MPC architecture coupled with the Wasserstein DRO scheme.

The MPC program specified in (48a)–(48i) details the slight modifications made to (47a)–(47d) accommodating the coupled PoE component to our LbC framework. We discuss this in more detail in Section IV-F.

One important note concerns a specific scenario of model adaptation where the true underlying system slowly changes. Our application of receding horizon control necessitates the use of a snapshot model in the prediction phase. This requires we assume the rate of change of the dynamics of the true plant is relatively small. In such conditions, however, the historical residuals we collect through measurements will slowly lose relevance. This issue can be easily reconciled with the use of either a moving window of residuals or with a proper forgetting scheme. In this article, we propose a simple method to accommodate such cases. Since the focus of this article is on *tabula-rasa* LbC, we relegate the discussion of this additional framework to this article's appendix.

E. Horizon Increment Rule

MPC with a well-defined dynamical structure can leverage judicious selection of the prediction horizon as a component to proving recursive feasibility. When considering a general class of systems as is the case with MBRL, the prediction horizon becomes a hyperparameter that manages the tradeoff between prediction depth and computational expense. In this article, we elect to define a simple horizon increment rule for our experiments. Typically, in LbC, the prediction horizon is a hyperparameter whose selection can be done empirically with more nuanced methods [37], [38]. In our case studies, which we design to emulate *tabula-rasa* LbC as closely as is consistent with the assumptions of our algorithm, we utilize this horizon increment rule as a heuristic to simply allow the problem to be rapidly solved. By solving severely restrictive case studies, we validate the performance of our method under the most challenging context for which it is technically designed. For real-world applications, the horizon can often be selected using a combination of available SME (which should not be ignored if it is available), and automatic tuning methods like those of [37] and [38]. The increment rule is not meant as a serious method for real-world embedded control systems that often possess highly limited computational resources.

F. PoE and Problem Assumptions

This section defines the set of least restrictive assumptions we identify toward achieving safe LbC. In this article, we consider systems with nonhybrid dynamics for simplicity. Our method leverages proved safety properties from [36], which apply to static optimization programs. We identify that these methods can apply to LbC problems under a series of assumptions made in this section. These assumptions almost entirely relate directly to situations when the dynamical, DRO, and PoE components, which are normally not considerations for static optimization programs, could create opportunities for empty feasible sets. This section defines a PoE scheme directly amenable to translating guarantees from [36] to our formulation. Notably, our assumptions are significantly less restrictive than those of existing LbC methods. The majority of these assumptions relate to clear necessary conditions which we detail here.

Assumption 1: A feasible state and control trajectory exists for each prediction horizon N in the optimal control problem. This is the most fundamental requirement to apply safe control.

Assumption 2: We assume we know a safe control input that we can apply at the first timestep.

Starting with limited model knowledge, if we do not know a temporarily safe control input we can apply at the first timestep, we obviously cannot translate any meaningful safety certificates. This contrasts with other work which requires knowledge of safe control trajectories throughout the time horizon or a known safe backup policy.

Assumption 3: Starting with an optimal control problem of the form (36a)–(36f), suppose we have a constraint function $g(x_k, u_k, \theta_{g;t}){:}\S \times \mathcal{U} \times \theta \rightarrow \mathcal{S}$. The sublevel set $\mathcal{G}_{r_{\mathrm{DRO}}} = \{(x, u) \in \S, \mathcal{U}{:}g(x, u) + r_{\mathrm{DRO}} \leq 0\}$ defines the adjusted feasible region, where feasibility is satisfied at the current timestep. This set must not be empty $\forall r_{\mathrm{DRO}} \in \mathcal{R}$, where the set $\mathcal{R} = \{r_{\mathrm{DRO}} \in \mathbb{R}{:}0 \leq r_{\mathrm{DRO}} \leq r_{\mathrm{DRO;max}}\}$ describes the set of all potential values of the DRO offset.

Since our method relies on creating an offset from the nominal constraint boundary, any potential value of the offset must lie in the image of the constraint function.

This assumption can be thought of as a generalization of a common LbC assumption that relates to "bounded modeling error," an example of which is given by Assumption 2 in [39]. In our case, using general function approximation, our method to quantify model error is empirically based on residuals. If the residuals of the learned model are too large, indicating our learned model is inaccurate, the resulting computed r_{DRO} (which is a conservative approximation of the residual, based on its distribution) will enforce a large offset from the nominal boundary. This assumption says that if the learned model is sufficiently inaccurate, the offset will be so large that the adjusted feasible region is empty, which is incompatible with the setup of [36]. The value $r_{\mathrm{DRO;max}}$ represents any maximum residual value we can potentially infer from the problem and can be defaulted to as an empirical approach if this case is reached in a real problem, although safety properties may not be reliable in such cases. Our experiments show such scenarios can be unlikely to occur, although the possibility of their occurrence should be considered.

The next assumption relates to a slightly stronger condition regarding PoE. The agent must be capable of exploring during LbC. To ensure the guarantees from [36] translate under those diverse circumstances, the same statements of 3.1–3.3 must be satisfied with respect to an additional exploration process $\mathcal{N}$ that ensures PoE.

For clarity, we define the following modified MPC program that considers an additive exploration signal from $\mathcal{N}$:

$$\min_{u, u^n \in \mathcal{U}} \sum_{k=t}^{t+N} J_k\left(\hat{x}_k, u_k\right) \tag{48a}$$

$$\text{s.t.: } \hat{x}_{k+1} = \hat{f}\left(\hat{x}_k, u_k, \theta_{g;t}\right) \tag{48b}$$

$$\hat{x}^n_{k+1} = \hat{f}\left(\hat{x}^n_k, u^n_k, \theta_{g;t}\right) \tag{48c}$$

$$\hat{g}\left(\hat{x}_k, u_k, \theta_{g;t}\right) + r_{\mathrm{DRO}} \leq 0 \tag{48d}$$

$$\hat{g}\left(\hat{x}_k^n, u_k^n, \theta_{g;t}\right) + r_{\text{DRO}} \leq 0 \tag{48e}$$

$$u^n = u + N_{i:i+N} \tag{48f}$$

$$N_{i:i+N} \sim \mathcal{N} \tag{48g}$$

$$\hat{x}_0 = x_t \tag{48h}$$

$$\hat{x}_0^n = x_t \tag{48i}$$

where $\mathcal{N}$ is the distribution of a random exploration process that can be added to the nominal control input, and the superscript x^n and u^n denote trajectories perturbed by the exploration signal. The solution $u^n(t)^\star$ is then applied to the plant at time step t.

Remark 4: Equations (48a)–(48i) guarantee feasibility from $k = t$ to $k = t + N$ for a system with parameters $\theta_{g;t}$ with a specified risk metric/probabilistic guarantee. This is formulated to guarantee feasibility over the control horizon. To assess recursive feasibility, one could utilize the methods from [19] and [20] that require more significant restrictions in the form of model knowledge, mathematical structure on the feedback policy, and prior existing safe data.

The additive noise perturbation for exploration takes inspiration from common methods with actor-critic or policy gradient learning, where noise via an Ornstein–Uhlenbeck process is added to the control input [40]. Relative to those existing methods, we make the following modifications for implementation:

Remark 5: We must constrain both nominal and perturbed trajectories to ensure safety even with exploration. If we only add the perturbation after solving the MPC program, safety is not guaranteed.

Remark 6: A scalarized tradeoff between $J_k(\hat{x}_k, u_k)$ and $J_k(\hat{x}_k^n, u_k^n)$ can be formulated to balance exploration and exploitation during planning.

Now, we define the next assumption relevant to translating safety to LbC systems under strong limitations on SME.

Assumption 4: Given the noise process $\mathcal{N}$ defined to satisfy PoE for the model identification problem, the constraints $g(x_k, u_k, \theta_{g;t})$ and $g(x_k^n, u_k^n, \theta_{g;t})$ of the snapshot model must be satisfied for every realization from $\mathcal{N}$ throughout the overall finite-time optimal control problem.

Given these conditions, we state the following remark detailing the properties of our method.

Remark 7: Based on the provided safety guarantee afforded by the adopted DRO framework from [36], (47a)–(47d) admits a feasible solution that satisfies the nominal constraints w.p. $1 - \eta$ as long as the feasible set is not empty, which follows from Assumptions 3.1–3.4.

We also state two remarks that help with the implementation of our approach.

Remark 8: These assumptions must also hold for the prediction horizons chosen at each instant in time.

Remark 9: If the DRO offset is so large, it creates an empty feasible set, and an artificial value $r_{\text{DRO;max}}$ can be defaulted to facilitate implementation, although safety guarantees in such situations may be difficult to translate. If a random search is used to solve the MPC program in such cases, the evaluated trajectory that creates the least predicted constraint violation given the unmodified DRO offset can be selected.

V. Case Study in Safe Online Lithium-Ion Battery Fast Charging

In this section, we validate our approach using a nonlinear lithium-ion battery fast charging problem. This problem closely emulates the performance-safety tradeoffs of common safe RL validation studies including ant-circle [41]. Specifically, the objective is to charge the battery cell as fast as possible, but the charging is limited by nonlinear voltage dynamics which must stay below critical thresholds. Violation of the voltage constraint can lead to rapid aging and potentially catastrophic failure. However, higher input currents (which increase voltage) also directly charge the battery more rapidly. Thus, the optimal solution is a boundary solution where the terminal voltage rides the constraint boundary. This presents a problem with significant challenges and tradeoffs relating to safety and performance. Exploring how such algorithms accommodate these challenges can reveal insights into their overall efficacy and shortcomings.

A. ECM of a Lithium-Ion Battery

Lithium-ion batteries can be modeled with varying degrees of complexity. Some of the more detailed dynamical models are based on electrochemistry. For example, the Doyle–Fuller–Newman (DFN) electrochemical battery model is a high-fidelity first-principles derived physics-based model of the dynamics within a lithium-ion battery [42]. Varying model-order reduction can be applied, yielding versions including the single particle model and the ECM. For simplicity, this article's case study utilizes an ECM. The relevant state variables in this model are the state of charge SOC and capacitor voltages V_{RC} in each of the two RC pairs. The relevant constraint is on the terminal voltage V. This constraint prevents the battery from overheating or aging rapidly during charging and discharging. The state evolution laws are given by

$$\text{SOC}_{k+1} = \text{SOC}_k + \frac{1}{Q} I_k \cdot \Delta t \tag{49}$$

$$V_{\text{RC}_1;k+1} = V_{\text{RC}_1;k} - \frac{\Delta t}{R_1 C_1} V_{\text{RC}_1;k} + \frac{\Delta t}{C_1} I_k \tag{50}$$

$$V_{\text{RC}_2;k+1} = V_{\text{RC}_2;k} - \frac{\Delta t}{R_2 C_2} V_{\text{RC}_2;k} + \frac{\Delta t}{C_2} I_k \tag{51}$$

$$V_k = V_{\text{ocv}}(\text{SOC}_k) + V_{\text{RC}_1;k} + V_{\text{RC}_2;k} + I_k R_0 \tag{52}$$

where $I(t)$ is the current input (which is the control variable for this problem), and V_{OCV} is the open-circuit voltage function, which is conventionally measured through experiments. The full experimental OCV curve is used to represent the true plant in the loop and is obtained from a lithium-iron phosphate (LFP) battery cell [43]. In this article, we learn the dynamics of the states and output using a simple feed-forward neural network.

B. Model-Predictive Control Formulation

We utilize the following formulation of fast charging:

$$\min_{I_k \in \mathcal{U}} \sum_{k=t}^{t+N} \left(\text{SOC}_k - \text{SOC}_{\text{target}}\right)^2 \tag{53}$$

TABLE I
SAFETY, COMPUTATIONAL, AND PERFORMANCE COMPARISON FOR DRO-MPC AND MPC WITH BATTERY FAST CHARGING. ACTIVATION OF THE DRO OFFSET BEGINS AT `minResidNum = 2`

(DRO)	% VIOLATIONS [%]	MAX VOLTAGE [V]	ITERATION TIME [S]	CHARGING TIME [MIN]
1	0.0 %	3.5944	0.8551	7.3833
2	0.4 %	3.7004	0.8473	7.7667
3	0.2 %	3.6887	0.8529	7.3000
4	0.6 %	3.7098	0.8503	8.1833
5	0.0 %	3.5927	0.8688	7.5333
6	0.4 %	3.7344	0.8550	7.7833
7	0.4 %	3.7032	0.8643	8.1167
8	0.2 %	3.6921	0.8692	7.6667
9	0.2 %	3.6916	0.8620	7.8667
10	0.2 %	3.6985	0.8375	8.0167
AVERAGES	0.26%	3.6806	0.8562	7.8150
RUN (NO DRO)	% VIOLATIONS [%]	MAX VOLTAGE [V]	ITERATION TIME [S]	CHARGING TIME [MIN]
1	4.2 %	3.7795	0.8630	6.8667
2	7.4 %	3.7604	0.8345	6.8667
3	5.0 %	3.7474	0.8055	6.7833
4	13.6 %	3.7284	0.7938	6.8500
5	8.0 %	3.9072	0.8020	6.8333
6	16.2%	3.9060	0.7977	6.8667
7	8.0 %	3.9040	0.8240	6.8667
8	11.6 %	3.7651	0.7875	7.0167
9	7.2 %	3.7736	0.8237	6.8000
10	16.4 %	3.7634	0.7928	6.7500
AVERAGES	9.76 %	3.8035	0.8125	6.8500

TABLE II
RELEVANT PARAMETERS FOR BATTERY CASE STUDY

PARAMETER	DESCRIPTION	VALUE	UNITS
Q	CHARGE CAPACITY	8280	$[\frac{1}{A.h}]$
R_0	RESISTANCE	0.01	$[\Omega]$
R_1	RESISTANCE	0.01	$[\Omega]$
R_2	RESISTANCE	0.02	$[\Omega]$
C_1	CAPACITANCE	2500	$[F]$
C_2	CAPACITANCE	70000	$[F]$
Δt	TIMESTEP	1	[S]
N_{targ}	MAX CONTROL HORIZON	8	[-]
η	RISK METRIC	0.025	[-]
β	AMBIGUITY METRIC	0.99	[-]
SOC_0	INITIAL SOC	0.2	[-]
SOC_{targ}	TARGET SOC	0.8	[-]
$V_{RC_1}(0)$	INIT. CAP. 1 VOLTAGE	0	[V]
$V_{RC_2}(0)$	INIT. CAP. 2 VOLTAGE	0	[V]

$$\text{s.t.: } (49) - (52), \quad \text{SOC}(0) = \text{SOC}_0 \tag{54}$$

$$V_k \leq 3.6 \text{ V}, \quad 0 \text{ A} \leq I_k \leq 40 \text{ A}. \tag{55}$$

The relevant parameters of the true model and DRO-MPC program are referenced in Table II.

Remark 10: In our case, we assume the controller does not have access to the form of the underlying dynamics given by (49) and (52). Instead, we apply our end-to-end LbC method to learn the dynamics "from scratch" as is consistent with *tabula-rasa* learning methods. We utilize neural network black-box models to accomplish this. The rules used to update the neural network parameters affect the convergence of the data-driven model to accurate behavior, which also effects empirical safety. We keep the neural network training consistent between our DRO algorithm and its nonrobust baseline. The exact training procedure can be referenced in [35]. Updating the model more slowly at first tends to encourage more consistent behavior.

In these case studies, we apply perturbation to the inputs that further excite the system, toward ensuring PoE. These perturbations are drawn as uniform vectors whose elements lie between $-2.5 \leq x_p \leq 2.5$ Amps. These perturbations are applied to both the distributionally robust controller, as well as the nonrobust baseline controller. In both cases, we seek to ensure mutual constraint satisfaction for the trajectories predicted using both the nominal and perturbed inputs.

We only allow a maximum total of 500 s for the battery to be charged. The timestep $\Delta t = 1$ s, $\eta = 0.025$, $\beta = 0.99$, and $N_{\text{targ}} = 8$ steps. Our neural network dynamical model has one hidden layer with three neurons and a sigmoid activation function, with a linear output layer. To solve the MPC problem, we apply a $(1 + \lambda)$ evolutionary strategy (ES) based on a normally distributed mutation vector. In our appendix, we describe how this strategy works, why we selected it, and other reasonable alternatives. The solver works with a single iteration and 250 000 mutants. The initial point of the ES is taken as the optimal point from the previous timestep. Addressing Assumption 2, we assume that at the first timestep, control inputs of $I_k \leq 25$ A are known to be temporarily safe. Since we constrain voltage which is a scalar, the constraint function dimension $m = 1$.

Our baseline is a learning MPC controller with no DRO framework. We adopt the same problem formulation as if we were going to add the constant r_{DRO} to the constraints, but we omit the DRO constant in the end to evaluate the impact it has on the robustness of the final control law.

C. Results

In total, we conducted ten experiments with identical designs but different initial random seeds. We run our algorithm and a nonrobust baseline for these ten runs on the

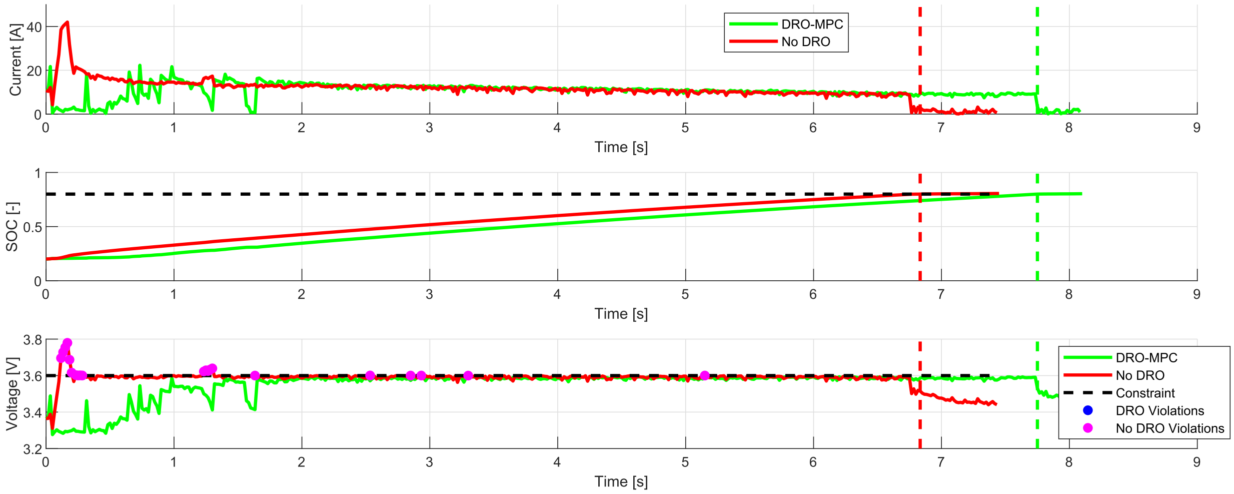

Fig. 2. Comparison of nonlinear MPC controller with and without DRO for lithium-ion battery fast charging. Run 1 is shown here.

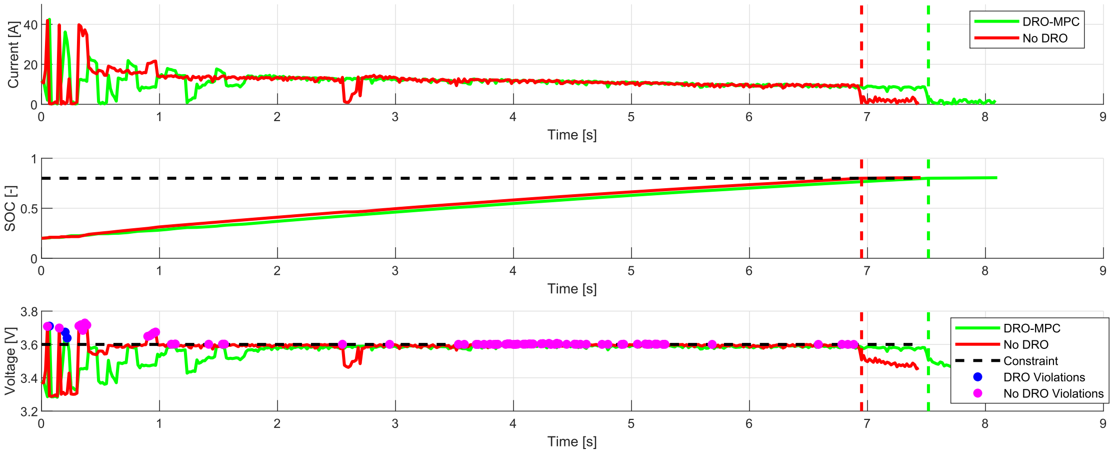

Fig. 3. Comparison of nonlinear MPC controller with and without DRO for lithium-ion battery fast charging. Run 4 is shown here.

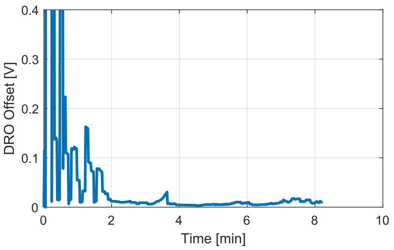

Fig. 4. Time evolution of DRO offset from run 4.

same battery fast charging problem detailed in the previous subsections. Table I shows the performance, computation, and safety statistics for each of these runs. For a closer look, we go to Fig. 2 that shows one run of both the DRO algorithm and its nonrobust counterpart. In the case of Fig. 2 (run 1), the DRO-based does not violate the constraint at any point. In Fig. 3, we see the highest incidence of constraint violation for the DRO controller (from run 4). Fig. 4 shows the time evolution of the DRO offset from run 4.

Conversely, the nonrobust versions both experience a combination of initial, significant voltage spikes, as well as minor violations that persist throughout the experiments. In total, if we focus on Fig. 3 (run 4), the nonrobust version violated constraints in 13.6 % of timesteps (68 timesteps out of 500 total). The charging time was 6.85 min, which was 16.29% faster than the DRO version, whose charging time was 8.1833 min. This makes intuitive sense, as the added DRO framework introduces additional conservatism which affects the performance of the overall control policy.

Overall, across all ten runs, our DRO version violates constraints in 0.26% of total timesteps, which is well within the chosen value of $\eta = 0.025 = 2.5\%$ over just a single optimization iteration. The nonrobust version, however, violates constraints in 9.76% of total timesteps on average. Similarly, there is a stark difference in the maximum voltages seen by the robust and nonrobust versions, with the DRO framework reducing the mean peak voltage by 122.9 mV. The DRO calculations increase the overall computation time by an average of 43.7 ms per timestep and allow the algorithm, in this case, to run in real-time. No optimizations were made to the MATLAB code to expedite the runtime of either algorithm, and the only difference in code between the two algorithms is the auxiliary and separate DRO framework. Finally, across the ten total runs, the overall charging time with the DRO framework averages 7.8150 min, approximately 14.1% longer than that of the non-DRO version. Given the safety-critical nature of this control problem, the safety guarantees of our algorithm are likely well worth the marginal degradation to the charging performance resulting from added conservatism.

VI. Case Study in Safe Autonomous Driving and Obstacle Avoidance

This section details the implementation of our algorithm for learning safe obstacle avoidance from scratch. This learning occurs within the same design as our battery case study, namely, we begin with zero model knowledge and only a single known safe control input. We fit a data-driven model to the dynamics and conduct receding-horizon control.

This study is designed with specific decisions in mind to more effectively reveal the efficacy of our algorithm. Some of these decisions make our study somewhat unrealistic insofar as they expose the agent to greater danger than necessary. Sections VI-A and VI-B provide discussion of these decisions in more detail.

A. Dynamical Simulator

In this case study, we utilize a bicycle model for the vehicle dynamics. This environment is encoded in the following equations discretized via forward Euler approximation:

$$x_{1;t+1} = x_{1;t} + \Delta t\big(x_{4;t}\cos\big(x_{3;t}\big)\big) \tag{56}$$

$$x_{2;t+1} = x_{2;t} + \Delta t\big(x_{4;t}\sin\big(x_{3;t}\big)\big) \tag{57}$$

$$x_{3;t+1} = x_{3;t} + \Delta t\left(x_{4;t}\frac{\tan\big(u_{2;t}\big)}{L}\right) \tag{58}$$

$$x_{4;t+1} = x_{4;t} + \Delta t\big(u_{1;t}\big) \tag{59}$$

where t is the current timestep, x_1 and x_2 are the x–y position of the vehicle, x_3 is the vehicle heading angle, x_4 is the vehicle velocity, u_1 is the acceleration input [in (m/s^2)], and u_2 is the steering angle input in radians. These equations represent the true plant, which is unknown to our learning-based controller.

B. Model Predictive Control Formulation

We utilize the following formulation of simple autonomous driving with obstacle avoidance:

$$\min_{u_k\in\mathcal{U}} -\left(x_1(t+N)+x_2(t+N)\right) \tag{60}$$

$$\text{s.t.: } (56)-(59), \quad x(0)=x(t) \tag{61}$$

$$Z(x_k) \le Z_{\text{cutoff}}, \quad u_{\min} \le u_k \le u_{\max}. \tag{62}$$

Here, $Z(x_k)$ is the obstacle function which represents a basic vision system. We limit Z to be smaller than a specified value (corresponding to the definition of the edge of the obstacle). Residuals in the DRO algorithm are with respect to this barrier using predicted values of the dynamical state, as opposed to the value of the obstacle function obtained with the true state. We create the environment defined by $Z(x_k)$ by generating and summing random Gaussians in two dimensions. Then, we define the obstacle boundaries by setting a threshold within the static map, below which becomes the safe region and above which the obstacles inhabit. This map is used with interpolation during the final experiment. If the constraint is violated, the agent will take actions that minimize the violation until feasibility is restored. We set $u_{\min} = [-1, -0.75]$ and $u_{\max} = -u_{\min}$. The experiment ends once the vehicle leaves the 100 × 100 m space.

TABLE III
Relevant Parameters for Obstacle Avoidance Case Study

Parameter	Description	Value	Units
L	Vehicle Length	0.5	[m]
Δt	Timestep	0.2	[s]
N_{targ}	Max Control Horizon	12	[-]
η	Risk Metric	0.005	[-]
β	Ambiguity Metric	0.99	[-]
$x_1(0)$	Initial x-position	5	[m]
$x_2(0)$	Initial y-position	10	[m]
$x_3(0)$	Initial vehicle angle	$\frac{\pi}{4}$	[rad]
$x_4(0)$	Initial velocity	0.5	[m/s]

With the learned neural network dynamics models, the MPC formulation in (60)–(62) becomes

$$\min_{u_k\in\mathcal{U}} -\big(\hat{x}_1(t+N)+\hat{x}_2(t+N)\big) \tag{63}$$

$$\text{s.t.: } \hat{x}_{k+1} = f^{NN}(x_k, u_k, \theta) \tag{64}$$

$$\hat{x}(0) = x(t) \tag{65}$$

$$Z\big(\hat{x}_k\big) \le Z_{\text{cutoff}} - r_{\text{DRO}} \tag{66}$$

$$u_{\min} \le u_k \le u_{\max}. \tag{67}$$

Table III includes relevant parameters of our case study design. In this case study, we simply use one-step residuals by relying on a basic assumption that the modeling error is uncorrelated to the depth of prediction. Based on our experiments, this assumption is reasonable.

We make a deliberate choice for this objective function for a host of reasons. While it necessarily encodes our intended behavior, it also is simple and at odds with the objective of avoiding obstacles. By allowing our simple objective function to drive the vehicle directly toward the obstacles, our control algorithm must be capable of managing the vehicle while simultaneously maintaining safety throughout the experiment. Thus, this case study is designed to specifically focus on the added safety contributions from the DRO framework.

For our learned model, we initialize a feed-forward neural network based on a single hidden layer with ten neurons. The hidden layer uses sigmoid activation functions, and the output layer uses linear activation. At the first timestep, we assume control inputs of a zero vector are known to be safe. To solve the MPC problem, we use the same $(1+\lambda)$ ES used in our battery case study. In this case, we modify the optimization algorithm such that we utilize 750 000 mutants. We also increase the maximum prediction horizon to $N_{\max} = 12$ to improve the consistency of our results.

C. Results

We conduct ten individual runs with both our algorithm and a nonrobust version. Figs. 5 and 6 show runs 1 and 3, respectively. Table IV shows safety statistics. With the DRO controller, only one of the ten total runs violates constraints at all and only during a single timestep. The overall violation with the DRO controller is 0.0623% of timesteps. Moreover, the magnitude of the violation with the DRO controller is

TABLE IV

SAFETY COMPARISON FOR DRO-MPC AND MPC WITH VEHICLE OBSTACLE AVOIDANCE. THE MAX VIOLATION IS IN TERMS OF THE EUCLIDEAN DISTANCE. THE NUMBERS IN PARENTHESES ARE THE TOTAL NUMBER OF TIMESTEPS WHERE CONSTRAINTS ARE VIOLATED, WITH THE DENOMINATOR BEING THE NUMBER OF TIMESTEPS BEFORE THE VEHICLE LEAVES THE 100 × 100 SIZED ENVIRONMENT

RUN	% VIOLATIONS (DRO)	MAX VIOLATION (DRO) [M]	% VIOLATIONS (NO DRO)	MAX VIOLATION (NO DRO) [M]
1	0% (0/156)	0	2.05 % (3/146)	0.3877
2	0 % (0/145)	0	0.65 % (1/155)	0.0121
3	0.57% (1/174)	0.0386	3.47 % (5/144)	0.4472
4	0 % (0/184)	0	7.94 % (17/214)	0.9986
5	0 % (0/167)	0	1.12 % (2/179)	0.1897
6	0 % (0/140)	0	8.55 % (23/269)	2.6259
7	0 % (0/148)	0	6.74 % (13/193)	1.6726
8	0 % (0/143)	0	4.73 % (8/169)	0.2581
9	0 % (0/182)	0	10.27 % (23/224)	1.1720
10	0 % (0/165)	0	1.14 % (2/175)	0.1772
AVERAGES	0.0623%	0.00386	5.193 %	0.8041

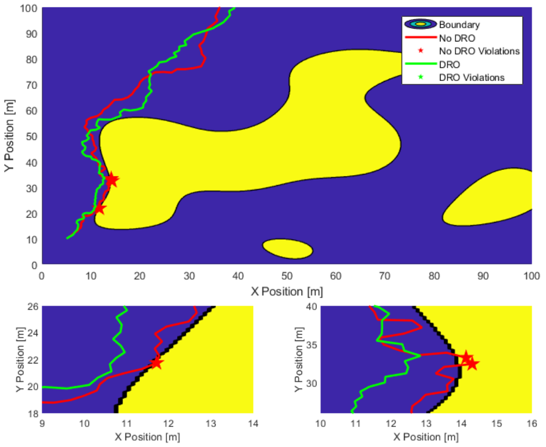

Fig. 5. Comparison of nonlinear MPC controller with and without DRO for vehicle obstacle avoidance. In this run, the DRO controller does not violate the constraints at all. This figure shows run 1, with the bottom plots revealing closeups of the areas with the highest constraint violation.

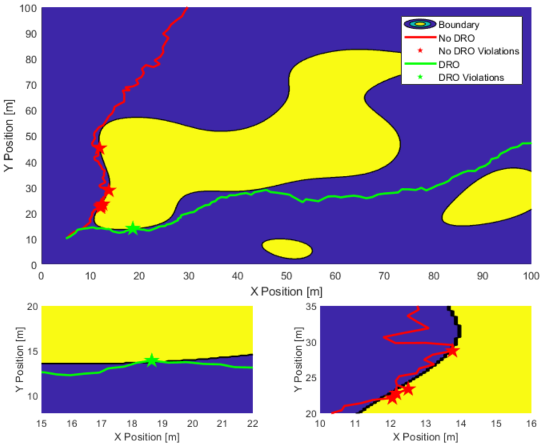

Fig. 6. Comparison of nonlinear MPC controller with and without DRO for vehicle obstacle avoidance. This figure shows run 3, with the bottom plots revealing closeups of the areas with the highest constraint violation.

equivalent to the vehicle skimming the edge of the boundary by less than 0.0386 m. Conversely, the nonrobust controller shows significant constraint violation in nearly all ten runs. The constraint violation of the nonrobust controller averages 0.8041 m of violation, which represents a complete collision with the obstacle (given our vehicle length $L = 0.5$). In one run, the nonrobust controller drives the vehicle nearly 3 m into the boundary before correcting and exiting the unsafe region. To verify the model is operating in nonlinear state space, Fig. 7 shows the range of the variable x_3 in run 1.

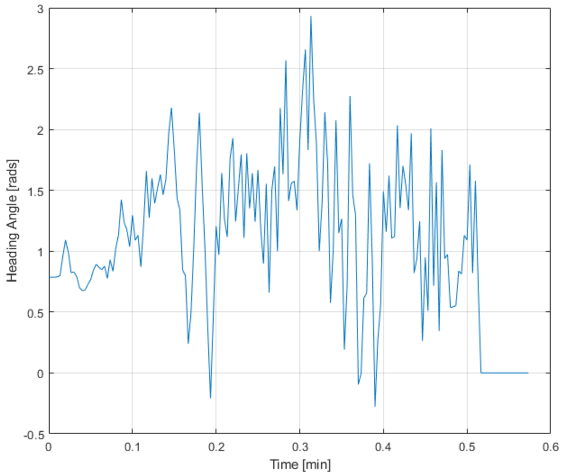

Fig. 7. Heading angle trajectory for run 1 (same as that shown in Fig. 6). The total range of heading angles is nearly π, showing exploration of highly nonlinear portions of the state space. The feasible range of steering angle input also covers a range of nonlinear behavior in the dynamics.

VII. DISCUSSION

Perhaps, the most important available insight is that for an application, the least amount of SME needed for synthesizing safe data-driven control is tied to the minimum amount of SME that yields a DRO offset that admits a feasible solution.

We have not only explored the behavior of our algorithm at the boundary of available knowledge and data but have validated its theoretical safety under a challenging arena of its applicability. Importantly, our approach is widely relevant in many LbC contexts (and for uncertainty quantification beyond control). For real-world applications, we are unlikely to conduct this restrictive type of *tabula-rasa* LbC. However, the same safety guarantees we have rigorously validated in these case studies are similarly applicable when more data and knowledge are available (e.g., conventional adaptive control, but with the modeling capacity of nonlinear machine-learning models). Since our approach functions as an end-to-end LbC method, it is also amenable to more unconventional applications including control synthesis from images or multimodal inputs [18]. We relegate exploration of this topic to future work.

VIII. Conclusion

This article presents an end-to-end distributionally robust model-based control algorithm. It addresses the problem of safety during LbC with strong limitations on our available knowledge and SME. We adopt a stochastic MPC formulation where we augment constraints with random variables corresponding to empirical distributions of modeling residuals. By applying Wasserstein ambiguity sets to optimize over the worst-case modeling error, we translate an out-of-sample safety guarantee subject to new data and experience. We validate this finding through simulation experiments. Our method applies to nonlinear MPC, but when applied to convex MPC programs it preserves convexity.

Appendix

A. Cardinality of Constraints Remains Constant

In the following lemma, we show that the number of constraints in the reformulation of the DRO problem in (46) need only be m (where m is the dimension of the constraint function output). When $g(\cdot)$ is nonseparable, as described in [36], then the number of constraints in the reformulation scales superlinearly as 2^m.

Lemma 3: If the modeling error residuals are defined as

$$R_1^{(t)} = \left|g\left(x_t, u_t^*\right) - \hat{g}\left(x_t, u_t^*, \theta_g\right)\right| \tag{68a}$$

$$R_1^{(t+1)} = \left|g\left(x_{t+1}^*, u_{t+1}^*\right) - \hat{g}\left(\hat{x}_{t+1}, u_{t+1}^*, \theta_g\right)\right| \tag{68b}$$

$$R_1^{(t+b)} = \left|g\left(x_{t+b}^*, u_{t+b}^*\right) - \hat{g}\left(\hat{x}_{t+b}, u_{t+b}^*, \theta_g\right)\right| \tag{68c}$$

and appear in the constraint function $g(\cdot)$ as (46), then the number of constraints in the reformulated problem remains identically m without jeopardizing the probabilistic guarantee.

Proof: Consider the following stochastic constraint converted to a DRCC:

$$x + \mathbf{R} \leq 0 \tag{69a}$$

$$\inf_{\mathbb{P}\in\mathbb{B}_\epsilon} \mathbb{P}[x + \mathbf{R} \leq 0] \geq 1 - \eta \tag{69b}$$

representing a constraint with uncertainty. Without loss of generality, we consider an MPC program with horizon $N = 1$.

The method of [36] enumerates across the vertices of a hypercube by modulating the sign of the DRO variable σ. However, when the random variable is a separable offset from a constant constraint boundary, we only need to consider perturbations that add conservatism. In the 1-D case, we can see from looking at the set of constraints

$$x \leq -r \quad \text{and } x \leq r \tag{70}$$

that only the first constraint $x \leq -r$ will ever be active. Therefore, $x \leq -r$ adequately defines the feasible region.

Likewise, if we consider the case where $\mathbf{R} \in \mathbb{R}^2$ with additive $\mathbf{R}$, we obtain the following set of constraints:

$$\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\end{bmatrix} + \begin{bmatrix}r_1\\ r_2\end{bmatrix} \leq 0 \tag{71}$$

$$\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\end{bmatrix} + \begin{bmatrix}-r_1\\ r_2\end{bmatrix} \leq 0 \tag{72}$$

$$\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\end{bmatrix} + \begin{bmatrix}r_1\\ -r_2\end{bmatrix} \leq 0 \tag{73}$$

$$\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\end{bmatrix} + \begin{bmatrix}-r_1\\ -r_2\end{bmatrix} \leq 0 \tag{74}$$

we see trivially that the feasible region defined by (71) and (74) is identical to that defined solely by (74). This pattern continues for any $m \in \mathbb{N}$ of $\mathbf{R} \in \mathbb{R}^m$.

B. Evolutionary Strategies and Random Search

In our article, we utilize a $(1+\lambda)$ ES to approximately solve the numerical MPC optimization program. This is a subset of what is generally referred to as a $((\mu/\rho) + \lambda)$ ES, whose precise definition can be referenced in [44]. A $((\mu/\rho) + \lambda)$ ES is a very simple form of a genetic algorithm, whereby at each generation/iteration of optimization, we have some number of "parents" who are mutated, and the parents are replaced by the highest performing mutated offspring. Random search is a highly effective method for solving optimization problems in RL literature [45]. Random search is also highly amenable to constrained optimization (without equality constraints), as infeasible mutants can be pruned from selection. If no feasible mutants are found, the mutant that least violates the constraint boundary can be selected to avoid additional computation.

C. Slow Model Adaptation

To accommodate potential cases where the true plant dynamics change slowly over time, we can adopt the following approach which preserves the safety guarantees of the Wasserstein DRO framework. We have system dynamics $x \in \mathbb{R}^n$ with no finite escape time. Furthermore, $g(x, u, \theta^*) \leq 0$ is our constraint function. Suppose it holds that the function g behaves in the following manner (similarly, although not identically, to a Lipschitz continuous function):

$$\max_{x\in\S, u\in\mathcal{U}, \delta\theta} |g(x, u, \theta + \delta\theta) - g(x, u, \theta)| \leq C \tag{75}$$

where $\delta\theta = \theta_{t+1}^* - \theta_t^*$ is any possible deviation in the model parameters throughout a single timestep. The value $\delta\theta$ is bounded. Consider we are at time t of the experiment. Let us represent the one-step residual at time $j = t - k$, where $k \in \{1, 2, \ldots, t\}$ is an integer, as

$$R_1^{(t)} = g\left(x_t, u_t, \theta_t^*\right) - \hat{g}(x_t, u_t, \theta_t) \tag{76}$$

where θ_t^* is the parameterization of the true plant at time t, and θ_t is the learned model at time t. If we add a value to the residual $R_1^{(t)}$ of $C \cdot k \cdot \mathrm{sgn}(R_1^{(t)})$

$$\tilde{R}_1^{(t)} = R_1^{(t)} + C \cdot k \cdot \mathrm{sgn}\left(R_1^{(t)}\right) \tag{77}$$

we accommodate for worst-case model adaptation in our algorithm. This scheme, coupled with a judiciously designed moving window of residuals, can accommodate model adaptation in the true underlying plant.

References

[1] K. J. Astrom, *Introduction to Stochastic Control Theory*. Chelmsford, MA, USA: Courier Corporation, 1970.

[2] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, Oct. 1996.

[3] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 3, no. 1, pp. 269–296, May 2020, doi: 10.1146/annurev-control-090419-075625.

[4] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *Proc. Amer. Control Conf. (ACC)*, Philadelphia, PA, USA, Jul. 2019, pp. 5582–5588.

[5] M. Bujarbaruah, X. Zhang, and F. Borrelli, "Adaptive MPC with chance constraints for FIR systems," 2018, *arXiv:1804.09790*.

[6] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari, "Adaptive receding horizon control for constrained MIMO systems," *Automatica*, vol. 50, no. 12, pp. 3019–3029, Dec. 2014.

[7] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. A data-driven control framework," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1883–1896, Jul. 2018.

[8] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," 2018, *arXiv:1803.08287*.

[9] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, Jul. 2019, pp. 3387–3395.

[10] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A.-A. Agha-mohammadi, and E. A. Theodorou, "Bayesian learning-based adaptive control for safety critical systems," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Aug. 2020, pp. 4093–4099.

[11] J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," 2020, *arXiv:2004.07584*.

[12] T. Westenbroek, A. Agrawal, F. Castaneda, S. Sastry, and K. Sreenath, "Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers," in *Proc. 7th IFAC Conf. Anal. Design Hybrid Syst.*, 2021, pp. 1–6.

[13] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 42, pp. 1437–1480, Aug. 2015.

[14] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-based approach to safe reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2018, pp. 1–10.

[15] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep RL with a safety critic," 2020, *arXiv:2010.14603*.

[16] J. Garcia and F. Fernandez, "Safe exploration of state and action spaces in reinforcement learning," 2014, *arXiv:1402.0560*.

[17] T. J. Perkins and A. G. Barto, "Lyapunov design for safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 3, pp. 803–832, Mar. 2003.

[18] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," 2015, *arXiv:1504.00702*.

[19] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3289–3304, Jul. 2022.

[20] Z. Zhong, E. A. del Rio-Chanona, and P. Petsagkourakis, "Data-driven distributionally robust MPC using the Wasserstein metric," 2021, *arXiv:2105.08414*.

[21] A. Nilim and L. El Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Oper. Res.*, vol. 53, no. 5, pp. 780–798, Oct. 2005.

[22] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, "Probabilistic safety constraints for learned high relative degree system dynamics," in *Proc. 2nd Conf. Learn. Dyn. Control*, 2020, pp. 781–792.

[23] B. P. G. Van Parys, D. Kuhn, P. J. Goulart, and M. Morari, "Distributionally robust control of constrained stochastic systems," *IEEE Trans. Autom. Control*, vol. 61, no. 2, pp. 430–442, Feb. 2016.

[24] J. A. Paulson, E. A. Buehler, and A. Mesbah, "Arbitrary polynomial chaos for uncertainty propagation of correlated random variables in dynamic systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3548–3553, Jul. 2017.

[25] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Math. Program.*, vol. 171, nos. 1–2, pp. 115–166, Sep. 2018.

[26] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with Wasserstein distance," 2016, *arXiv:1604.02199*.

[27] C. Zhao and Y. Guan, "Data-driven risk-averse stochastic optimization with Wasserstein metric," *Oper. Res. Lett.*, vol. 46, no. 2, pp. 262–267, Mar. 2018.

[28] I. Yang, "Wasserstein distributionally robust stochastic control: A data-driven approach," 2018, *arXiv:1812.09808*.

[29] A. Kandel, S. Park, and S. J. Moura, "Distributionally robust surrogate optimal control for high-dimensional systems," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 3, pp. 1196–1207, May 2023.

[30] A. Kandel and S. J. Moura, "Safe Wasserstein constrained deep Q-learning," 2020, *arXiv:2002.03016*.

[31] E. Lecarpentier and E. Rachelson, "Non-stationary Markov decision processes a worst-case approach using model-based reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst* Red Hook, NY, USA: Curran Associates, 2019, pp. 7216–7225.

[32] K. Asadi, D. Misra, and M. L. Littman, "Lipschitz continuity in model-based reinforcement learning," 2018, *arXiv:1804.07193*.

[33] I. Akbar, "Uncertainty estimation in continuous models applied to reinforcement learning," Ph.D. dissertation, Dept. Elect. Eng. (Intell. Syst., Robot., Control), Univ. California San Diego, La Jolla, CA, USA, 2019.

[34] I. Yang, "A convex optimization approach to distributionally robust Markov decision processes with Wasserstein distance," *IEEE Control Syst. Lett.*, vol. 1, no. 1, pp. 164–169, Jul. 2017.

[35] A. Kandel. (Aug. 2023). *Wasserstein Nonlinear MPC*. [Online]. Available: https://github.com/aaronkandel/Wasserstein-Nonlinear-MPC/tree/main

[36] C. Duan, W. Fang, L. Jiang, L. Yao, and J. Liu, "Distributionally robust chance-constrained approximate AC-OPF with Wasserstein metric," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4924–4936, Sep. 2018.

[37] B. Zhang et al., "On the importance of hyperparameter optimization for model-based reinforcement learning," 2021, *arXiv:2102.13651*.

[38] L. Li, K. G. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Efficient hyperparameter optimization and infinitely many armed bandits," *arXiv:1603.06560*, 2016.

[39] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon et al., Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 1–11.

[40] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[41] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, NSW, Australia, 2017, pp. 1–10.

[42] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *J. Electrochem. Soc.*, vol. 140, no. 6, pp. 1526–1533, Jun. 1993.

[43] H. E. Perez, X. Hu, S. Dey, and S. J. Moura, "Optimal charging of Li-ion batteries with coupled electro-thermal-aging dynamics," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7761–7770, Sep. 2017.

[44] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, Mar. 2002.

[45] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," 2018, *arXiv:1803.07055*.