2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Scalable Estimation of Invariant Sets for Mixed-Integer Nonlinear Systems using Active Deep Learning

Angelo D. Bonzanini[1], Joel A. Paulson[2], Georgios Makrygiorgos[1], Ali Mesbah[1]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993131

Abstract— **Set invariance is a crucial property for ensuring safe and feasible performance of closed-loop systems under state and input constraints. Classical set-theoretic methods for constructing reachable and invariant sets are generally inadequate in handling complex system dynamics and may not be scalable to high-dimensional systems. This paper presents a sample-efficient approach for data-driven estimation of invariant sets for constrained nonlinear systems that can exhibit a mixture of continuous, discrete, and/or switching-mode behavior. The approach relies on learning an oracle that verifies if a given system state is feasible. Thus, the invariant set construction problem is converted to a classification problem that can be effectively solved with deep learning. We also present an active learning algorithm to improve the sample efficiency of deep learning-based estimation of the feasibility oracle. Randomized verification is then used to provide probabilistic guarantees for set invariance. The proposed approach does not impose any assumptions on the structure of system dynamics, and is particularly suitable when the feasibility test for control invariance requires solving (expensive) mixed-integer nonlinear programs. The approach is illustrated on a benchmark problem.**

I. Introduction

Set invariance theory is fundamental to the design and control of constrained systems [1]. This stems from the fact that system constraints can be satisfied at all times if and only if the initial state of a system lies inside an invariant set. As such, the notion of set invariance is essential for ensuring *safe* evolution of a dynamical system under some admissible input sequence. Recent years have witnessed a renewed interest in the theory and application of set invariance; for example, in the areas of predictive and learning-based control of uncertain systems to guarantee feasibility and constraint satisfaction, thereby ensuring safe learning and control of dynamical systems (e.g., [2], [3], [4]).

Methods for constructing invariant sets based on set operations are well-established for linear time-invariant systems subject to linear inequality constraints [5], [6], [7]. Although these methods cannot be used for general nonlinear systems, algorithms have been developed for constructing reachable sets and control invariant sets, which are generally harder to compute, for certain classes of nonlinear (such as piecewise affine) and hybrid systems [8], [9], [10], [11], [12]. Nonetheless, construction of these sets for general dynamical systems that exhibit complex forms of nonlinearities and/or a mixture of continuous and discrete behavior, as commonly observed in hybrid and switched systems, remains an open problem. Not only is the construction of reachable and invariant sets using set-theoretic methods strongly dependent on the structure and accuracy of a system model, scalability of these methods is also an important limitation since set-based operations can quickly become prohibitively expensive for high-dimensional state and input spaces [13].

Learning-based methods have recently been proposed for data-driven approximation of reachable and invariant sets, as an alternative to their explicit construction. The core idea of these methods is to generate state trajectories in a data-driven manner by simulating a complex system from some initial state sampled from an admissible state-space. These samples can then be used to estimate reachable/invariant sets using the state-of-the-art machine learning approaches. A key advantage of learning-based methods for set construction is that they can even be applied when the underlying system dynamics are completely unknown; for example, when only a black-box or high-fidelity system model is available. However, their performance strongly depends on the sampling of the state-space. To this end, an active learning method is proposed in [14] to intelligently select batches of samples that are most informative and least redundant to previously labeled samples via submodular maximization. In [15], a deep learning approach is presented to compute reachable sets using the Hamilton-Jacobi-Isaacs variational inequality, while formulating the value function as a neural network that is trained using a "physics-informed" loss function. An approach for the computation of reachable sets for high-dimensional nonlinear systems is proposed in [16], where sets are represented as zonotopes coupled with local linearization whose error is captured by Lagrange remainders.

This paper presents a sample-efficient, learning-based approach for estimating invariant set representations for general constrained nonlinear systems with mixtures of discrete states, inputs, and/or operating modes. The proposed approach circumvents the need for making any assumptions about the structure and complexity of system dynamics. It relies on the notion of learning a *feasibility oracle* to determine whether a given system state lies within some invariant set. By repeatedly querying this oracle for different system states, the invariant set construction problem is transformed from a set-based reachability problem into a binary classification problem that can be tackled with deep learning tools [17], which are highly versatile in learning complex (non-convex and/or disjoint) classification boundaries. To

[1] A. D. Bonzanini, G. Makrygiorgos, and A. Mesbah are with the Department of Chemical and Biomolecular Engineering at the University of California, Berkeley, CA 94720.

[2] Joel A. Paulson is with the Department of Chemical and Biomolecular Engineering at the Ohio State University, Columbus, OH 43210.

This work was supported in part by the National Science Foundation Grant 1839527.

enhance the sample efficiency of the oracle estimation via deep learning, we present an active learning algorithm based on information-theoretic criteria to sequentially enrich training data in an optimal manner. Active learning allows us to mitigate the computational cost of invoking the oracle repeatedly on randomly-selected samples of the state, which can be prohibitive when the feasibility test for control invariance involves solving mixed-integer nonlinear programs. To enhance the overall performance of the set approximation, we leverage randomized verification methods [18] to provide a probabilistic guarantee on the likelihood that a state within the learned invariant set boundary is truly invariant; these probability metrics are also utilized as stopping criteria for the supervised learning task, hence reinforcing the sampling efficiency. The proposed approach is demonstrated on a benchmark two-tank system that exhibits switching behavior.

II. Problem Statement

Consider a general representation of nonlinear systems that can involve a mixture of continuous and discrete states and control inputs

$$x_c^+ = f_c(x_c, x_d, u_c, u_d), \tag{1a}$$
$$x_d^+ = f_d(x_c, x_d, u_c, u_d), \tag{1b}$$

where $x_c \in \mathbb{R}^{n_{x_c}}$ and $u_c \in \mathbb{R}^{n_{u_c}}$ are the continuous state and control input, respectively; $x_d \in \mathbb{Z}^{n_{x_d}}$ and $u_d \in \mathbb{Z}^{n_{u_d}}$ are the discrete state and control input, respectively; and $f_c : \mathbb{R}^{n_{x_c}} \times \mathbb{Z}^{n_{x_d}} \times \mathbb{R}^{n_{u_c}} \times \mathbb{Z}^{n_{u_d}} \to \mathbb{R}^{n_{x_c}}$ and $f_d : \mathbb{R}^{n_{x_c}} \times \mathbb{Z}^{n_{x_d}} \times \mathbb{R}^{n_{u_c}} \times \mathbb{Z}^{n_{u_d}} \to \mathbb{Z}^{n_{x_d}}$ represent the continuous and discrete system dynamics, respectively. In the remainder of this paper, we compactly denote the general nonlinear system (1) by

$$x^+ = f(x, u), \tag{2}$$

where $x = \{x_c,\ x_d\}$ and $u = \{u_c,\ u_d\}$ are the collection of discrete states and control inputs, respectively. The system (2) is assumed to be subject to pointwise-in-time constraints on both the input and state

$$u \in \mathbb{U}, \quad x \in \mathbb{X}. \tag{3}$$

We now recall concepts of controllable and control invariant sets that are central to the feedback control of constrained systems (e.g., see [6], [7]).

Definition 1 (T-step controllable set). The *T-step controllable set* $\mathcal{K}_T(\Omega, \mathcal{T})$ is the largest set of initial states x_0 for which there exists an admissible control input sequence $\{u_0, \ldots, u_{T-1}\}$ such that an arbitrary terminal set $\mathcal{T} \subset \Omega$ is reached in exactly T steps while keeping the evolution of the state inside the set Ω for the first $T-1$ steps, i.e.,

$$\mathcal{K}_T(\Omega, \mathcal{T}) = \{x_0 \,|\, \exists \{u_k \in \mathbb{U}\}_0^{T-1} : \{x_k \in \Omega\}_0^{T-1}, x_T \in \mathcal{T}\}.$$

The controllable sets for the system (2) can be computed recursively as

$$\begin{aligned} \mathcal{K}_0(\Omega, \mathcal{T}) &= \mathcal{T}, \\ \mathcal{K}_{i+1}(\Omega, \mathcal{T}) &= \mathcal{Q}\big(\mathcal{K}_i(\Omega, \mathcal{T})\big) \cap \Omega, \end{aligned} \tag{4}$$

where $\mathcal{Q}(\cdot)$ denotes the *one-step set* (also known as the precursor or one-step backward reachable set), i.e.,

$$\mathcal{Q}(\Omega) = \{x_k \,|\, \exists u_k \in \mathbb{U} : f(x_k, u_k) \in \Omega\}.$$

It follows that the main step in computing the controllable sets is construction of the one-step set via evaluating the precursor set operation $\mathcal{Q}$. While $\mathcal{Q}$ can be straightforwardly evaluated for (small-dimensional) linear systems subject to polytopic constraints, for general nonlinear systems, especially those with a mixture of continuous and discrete state and input, evaluating $\mathcal{Q}$ is challenging or even completely intractable even using state-of-the-art set theoretic tools.

We now define the notion of control invariant sets that allows us to determine whether, given an initial state, it is possible to choose a feasible control input sequence such that the state evolution satisfies state constraints for all times.

Definition 2 (Control invariant set). The set Ω is a *control invariant set* for the system (2) if and only if there exists a control input $u_k \in \mathbb{U}$ such that the following condition holds

$$x_k \in \Omega \Rightarrow \exists u_k \in \mathbb{U} : f(x_k, u_k) \in \Omega, \tag{5}$$

for all $x_k \in \Omega$, i.e., $\exists u \in \mathbb{U} : f(\Omega, u) \subseteq \Omega$.

One is often interested in constructing the largest control invariant set contained in some set Ω.

Definition 3 (Maximal control invariant set). The set $\mathcal{C}_\infty(\Omega)$ is the *maximal control invariant set* contained in Ω for the system (2) if and only if $\mathcal{C}_\infty(\Omega)$ is control invariant and contains all the control invariant sets contained in Ω.

Control invariant set theory plays a fundamental role in the control of the constrained system (2). According to Definitions 2 and 3, there exists an admissible control law for (2) such that the state constraints in (3) can be satisfied for all $k \in \mathbb{N}$ if and only if the initial state $x_0 \in \mathcal{C}_\infty(\mathbb{X}) \subseteq \mathbb{X}$. Most algorithms for testing whether Ω is control invariant are based on the well-known *geometric condition for invariance* [5], which states that the set Ω is a control invariant set if and only if $\Omega \subseteq \mathcal{Q}(\Omega)$. As such, testing for control invariance relies on constructing the one-step set $\mathcal{Q}(\Omega)$, which is nontrivial for general dynamical systems, as described above.

Here, we present an approach for data-driven approximation of control invariant sets for the general nonlinear system (2), subject to constraints (3), when a control input sequence $\{u_0, \ldots, u_{T-1}\}$ is determined by repeatedly solving a dynamic optimization problem that provides a feasibility test for the control invariance condition. For the general setting considered here, this optimization problem belongs to the challenging class of mixed-integer nonlinear programs (MINLPs). MINLPs are increasingly used for optimization and control of complex dynamical systems with mixtures of discrete states, inputs, and/or discrete operating modes, wherein ensuring feasibility is crucial; for example, in safety-critical applications. The proposed approach circumvents the need for direct computation of invariant sets using set-theoretic methods [1], which is currently impractical for the system (2). As described next, the core idea of the proposed approach is to learn an oracle, using data generated from solving a (typically computationally expensive) MINLP, to determine whether a given system state is feasible for all future time steps. Accordingly, the invariant set construction problem is transformed into a binary classification problem by repeatedly querying the oracle for randomly-selected values of the state. Section IV presents an active learning algorithm for sample-efficient querying of the oracle, while providing probabilistic guarantees that a state within the learned invariant set satisfies the invariance condition (5).

III. Deep Neural Network Learning-based Framework for Invariance

To transform the control invariant set construction problem into a "learning" problem, we make two key observations. First, based on the definition of a T-step controllable set, we can construct a control invariant set $\mathcal{C} := \mathcal{K}_T(\mathbb{X}, \mathbb{X}_f)$ by running the algorithm in (4) with $\Omega = \mathbb{X}$ and $\mathcal{T} = \mathbb{X}_f$ for any $T \geq 1$, where $\mathbb{X}_f \subset \mathbb{X}$ is any control invariant set for (2) that is contained within the state constraints. This was proved in [19, Theorem 2], along with the fact that each set contains the previous one, i.e., $\mathcal{K}_T(\mathbb{X}, \mathbb{X}_f) \subseteq \mathcal{K}_{T+1}(\mathbb{X}, \mathbb{X}_f)$. Second, we can test if any particular initial state x_0 satisfies the *finite* set of constraints defining $\mathcal{C}$ by solving the following parametric feasibility problem for any fixed parameter $x_0 \in \mathbb{X}$

$$\begin{aligned} \text{find} \quad & \{u_0, \ldots, u_{T-1}\}, && \text{(6a)} \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, \ldots, T-1\}, && \text{(6b)} \\ & (x_k, u_k) \in \mathbb{X} \times \mathbb{U}, \quad \forall k \in \{0, \ldots, T-1\}, && \text{(6c)} \\ & x_T \in \mathbb{X}_f. && \text{(6d)} \end{aligned}$$

This problem can be straightforwardly cast as an (MIN)LP depending on the structure of the dynamics f and the constraints $\mathbb{X}$ and $\mathbb{U}$, e.g., by formulating it as a minimization problem with a zero-value objective function. It is important to note the breadth of problems that can be represented by (6). When the dynamics are linear and the constraints are polytopes, (6) will reduce to a simple convex linear program that can be solved efficiently to global optimality (even when the state, input, or horizon T are very large). When the constraints involve discrete variables (e.g., due to logical conditions), but the dynamics remain linear, then (6) becomes an MILP for which significant algorithmic advances have been made in the past decade so that they are applicable to large-scale problems. When the dynamics and/or the constraints include nonlinear terms, then we must rely on state-of-the-art global MINLP solvers.

Additionally, the choice of $\mathbb{X}_f$ is flexible in the sense that $\mathcal{C}$ will be control invariant for *any* control invariant $\mathbb{X}_f$. A significant amount of work has been done on the construction of inner control invariant approximations to the maximal control invariant set for certain classes of linear and nonlinear systems (see, e.g., [1], [7], [9], [20]). Even for the general system (2), we can always select $\mathbb{X}_f = \{x_{ss}\}$, where $x_{ss} \in \mathbb{X}$ denotes a feasible steady-state value that satisfies $x_{ss} = f(x_{ss}, u_{ss})$ for some $u_{ss} \in \mathbb{U}$. This is not a restrictive assumption in practice since most relevant engineering and control systems are designed to have at least one safe operating mode. Therefore, the main value of (6) is that it allows us to systematically enlarge a known "safe" set $\mathbb{X}_f$, which is strictly non-decreasing for increasing T.

If we can find a feasible solution to (6) for any fixed value of x_0, we can then guarantee that $x_0 \in \mathcal{C}$. Furthermore, if we can prove that no feasible value exists (through a guaranteed global solver), then we know that $x_0 \notin \mathcal{C}$. As such, a *feasibility oracle* can be defined as the following function that maps initial states to a binary number

$$\mathcal{O}(x) = \begin{cases} +1, & \text{if (6) is feasible for } x_0 = x, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Since we must have $\mathcal{O}(x) = 1, \forall x \in \mathcal{C}$ and $\mathcal{O}(x) = 0, \forall x \notin \mathcal{C}$, we can interpret $\mathcal{C}$ as the decision boundary of the classification oracle $\mathcal{O} : \mathbb{X} \to \{0, 1\}$. Thus, we have transformed the set construction problem to that of learning the function $\mathcal{O}$. The main advantage of this new view is that we can query the oracle to generate data of the form

$$\mathcal{D} = \{(x_1, \mathcal{O}(x_1)), (x_2, \mathcal{O}(x_2)), \ldots, (x_q, \mathcal{O}(x_q))\}, \tag{8}$$

where q denotes the number of samples. Since we do not know the structure of $\mathcal{O}$, which may be complex, deep neural networks (DNNs) are used to learn this mapping. DNNs with L hidden layers are postulated to be universal function approximators under relatively mild conditions [17], given a suitable selection of activation functions. Here, the activation function in the output layer is a sigmoid function, which ensures that the DNN outputs an estimated probability of the feasible class label $+1$ [17]. In the remainder, the DNN feasibility oracle is represented by $\mathcal{M}(x, \theta)$, where θ denotes the DNN parameters (i.e., weights and biases).

The DNN parameters are trained by minimizing a standard binary classification loss function

$$\mathcal{L}(\theta) = \frac{1}{N_t} \sum_{i=1}^{N_t} y_i \log(p_i(\theta)) + (1 - y_i) \log(1 - p_i(\theta)),$$

where y_i and $p_i(\theta) = \mathcal{M}(x_i; \theta)$ denote, respectively, the true probability and the predicted probability of the true/feasible class (+1) for the i^{th} training sample. In binary classification, a probability threshold is used to assign each new observation with a specific class. Typically, a value of $\pi_{\text{th}} = 0.5$ is used to distinguish between the two classes, though this can be tuned as discussed in the next section. In practice, the number of nodes, layers, and internal activation functions constitute hyperparameters of a DNN. Efficient tools, such as Bayesian optimization [21], can be used to automate the hyperparameter selection procedure.

We assumed the existence of a sufficiently rich dataset in (8) to learn an accurate decision boundary. However, since $\mathcal{C}$ may have a complex representation, a large number of samples may be needed to achieve an accurate model of the decision boundary, especially if the samples are naively chosen. This would not be a major issue if querying $\mathcal{O}$ was cheap; however, it can become prohibitively expensive in this work since performing the feasibility test generally requires the global solution to an MINLP problem. Therefore, we must select samples $\{x_1, \ldots, x_{N_t}\}$ such that the most informative labeled output values are produced. We present a strategy to identify these informative samples next.

IV. Scalable Learning-based Design of Control Invariant Sets

Section III presented a general methodology for learning control invariant sets by fitting DNN approximators of the oracle function $\mathcal{O}$ defined in (7). However, we did not address two major practical issues with such a framework: (i) the selection of the training dataset $\mathcal{D}$, which is expensive to construct; and (ii) systematic verification that the control invariance condition holds despite approximation errors. We address both of these challenges by first developing an active learning algorithm to sequentially populate $\mathcal{D}$ with informative samples and then proposing a probabilistic validation method that can provide statistical guarantees that the control invariance condition holds for the approximated set.

A. Active learning for DNN-based classifiers

To limit the number of required training samples as much as possible, we employ active learning (AL), which is a special class of supervised learning algorithms that interactively query the oracle to label new samples in a sequential fashion [22], [23]. Although there has been a significant amount of work on AL, we focus on *pool-based sampling* [24], which assumes that there exists a small set of labeled (input-output) data denoted by $\mathcal{D}$ and a large pool of unlabeled (input only) data denoted by $\mathcal{P}$. The unlabeled dataset is queried in a greedy fashion using some metric of informativeness of each sample in $\mathcal{P}$ [22]. The most commonly used query framework is uncertainty sampling, where the learning algorithm queries the samples that are deemed to be the most uncertain. Common types of uncertainty sampling include least confidence [24], margin sampling [25], and information entropy [26]. A conceptual representation of AL is shown in Fig. 1.

The choice of stopping criterion can have a significant impact on the overall performance of an AL algorithm. For example, if one specifies a maximum budget for querying the oracle, the learned decision boundary may not be very accurate since we do not know how large to set this value *a priori*. To this end, we propose a novel stopping criterion for the invariant set learning problem based on geometric arguments. The main idea is to stop the AL algorithm once the change in the feasible hyper-volume of the predicted control invariant set between two consecutive iterations falls below a specified tolerance. Since the feasible area is a function of the probability threshold π_{th}, we also require the sensitivity of the hyper-volume to be low, i.e., the area difference is small for a range of π_{th}. Let $V_k(\pi)$ denote the hyper-volume of the predicted control invariant set $\mathcal{M}(x;\theta) \geq \pi$ for a fixed threshold π. We formulate our proposed error metric as

$$\varepsilon = \frac{1}{\pi^U - \pi^L} \int_{\pi^L}^{\pi^U} |V_k(\pi) - V_{k-1}(\pi)| d\pi, \qquad (9)$$

for chosen bounds on the probability threshold π^L and π^U. Note that, in practice, we discretize the integral in (9) and use a large set of random samples to approximate the hyper-volume of the approximated control invariant set. This can often be done efficiently via batch processing of input samples to the DNN.

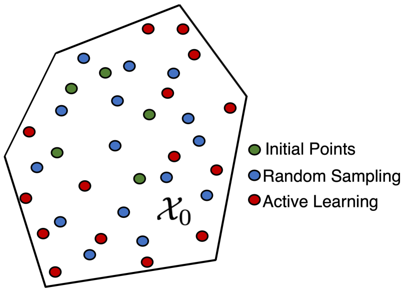

Fig. 1. Conceptual representation of active learning compared to random sampling. Initially, a few samples (green circles) are chosen from the set of initial conditions. Random search queries samples drawn randomly from the set, while an active learning algorithm selects samples according to their information content (e.g., samples closer to boundaries or away from initial/previous samples are more informative).

Algorithm 1 Proposed active learning algorithm for scalable construction of control invariant sets

```
Require: Initial labeled dataset $\mathcal{D}_0$, pool of unlabeled samples $\mathcal{P}$, feasibility oracle $\mathcal{O}$, information metric $\mathcal{I}$, number of samples to add at each iteration $S$, and stopping threshold $\varepsilon_{\text{tol}}$.
Train DNN model $\mathcal{M}_0(x)$ using $\mathcal{D}_0$ (Section III).
Set iteration counter $j = 0$.
Set $\varepsilon = +\infty$.
while $\varepsilon \geq \varepsilon_{\text{tol}}$ do
    Evaluate $I_j = \mathcal{I}(\mathcal{M}_0, \mathcal{P})$.
    Set $\mathcal{X}_j$ to be $S$ best values of $x$ in list $I_j$.
    Compute labels for chosen points $\mathcal{Y}_j = \mathcal{O}(\mathcal{X}_j)$.
    Add new labeled points to the dataset $\mathcal{D}_{j+1} = \mathcal{D}_j \cup \{(\mathcal{X}_j, \mathcal{Y}_j)\}$ and delete $\mathcal{X}_j$ from pool $\mathcal{P}$.
    Train new DNN $\mathcal{M}_{j+1}(x)$ using $\mathcal{D}_{j+1}$ (Section III).
    Calculate the current error metric $\varepsilon$ using (9).
    Increment iteration counter $j \leftarrow j + 1$.
end while
```

The AL algorithm with the proposed stopping criterion is summarized in Algorithm 1. This algorithm requires the user to specify an information metric $\mathcal{I}(\mathcal{M}, x)$ that maps a model $\mathcal{M}$ and a proposed point x to a real number. The entropy function is a commonly chosen representation of $\mathcal{I}$, which is defined for a binary classification problem as

$$\begin{aligned} \mathcal{I}(\mathcal{M}, x) = &-\mathcal{M}(x)\log(\mathcal{M}(x)) \\ &-(1-\mathcal{M}(x))\log(1-\mathcal{M}(x)), \end{aligned} \qquad (10)$$

where $\mathcal{M}(x)$ is the probability of the positive class (+1) being predicted and $1 - \mathcal{M}(x)$ is the probability of the negative class (0) being predicted. From (10), we see that the information content goes to 0 as the probability of belonging to the positive class approaches 0 or 1. This reflects our intuition that the information content will be largest for values that are nearest to the current decision boundary.

B. Probabilistic verification of control invariance condition

Even though our proposed AL approach (Algorithm 1) can greatly improve upon traditional random sampling methods, it is not guaranteed that the algorithm returns a control invariant set in light of the approximation errors incurred during training. Let $\tilde{\mathcal{C}}$ denote the approximated control invariant set computed as a level set of the DNN returned by Algorithm 1. As discussed earlier, one way to test that this set is control invariant is to determine if $\tilde{\mathcal{C}} \subseteq \mathcal{Q}(\tilde{\mathcal{C}})$. Instead of trying to exactly check this condition, we develop a one-step feasibility test

$$\phi(x) = \begin{cases} 0, & \text{if } \exists u \in \mathbb{U} \text{ such that } f(x,u) \in \tilde{\mathcal{C}} \\ 1, & \text{otherwise.} \end{cases} \tag{11}$$

The control invariance test can then be mathematically stated as $\phi(x; \tilde{\mathcal{C}}) = 0$ for all $x \in \tilde{\mathcal{C}}$, which is equivalent to the following maximization problem $\max_{x \in \tilde{\mathcal{C}}} \phi(x) = 0$. Since this is a very challenging problem to solve directly, we resort to computing a probabilistic estimate by associating a probability measure $\text{Pr}_{\tilde{\mathcal{C}}}$ over the sample space $\tilde{\mathcal{C}}$. We can then compute an empirical maximum over a set of N i.i.d. samples of $x \sim \text{Pr}_{\tilde{\mathcal{C}}}$ as follows

$$\hat{E}_N = \max_{i=1,\dots,N} \phi(x^{(i)}). \tag{12}$$

We can apply seminal results from [27] to this probabilistic estimator to establish an explicit bound on the number of samples required to achieve a desired level of accuracy. In particular, [27, Theorem 3.1] proves that if

$$N \geq \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}}, \tag{13}$$

then

$$\text{Pr}_{\tilde{\mathcal{C}}^N} \left\{ \text{Pr}_{\tilde{\mathcal{C}}} \{ \phi(x) \geq \hat{E}_N \} \leq \epsilon \right\} \geq 1 - \delta, \tag{14}$$

where $\epsilon \in (0,1)$ is often referred to as the *accuracy* and $\delta \in (0,1)$ as the *confidence*. It is interesting to note that the bound in (13) is completely independent of the size of the set $\tilde{\mathcal{C}}$ and the type of probability measure $\text{Pr}_{\tilde{\mathcal{C}}}$. Also, note that there are two levels of probabilities in (14) since $\hat{E}_N$ is a random variable that depends on the multi-sample $\{x^{(1)}, \dots, x^{(N)}\}$. As such, the scenario bound (14) can be interpreted as ensuring that the set of points greater than the estimated worst-case binary feasibility value has a measure smaller than ϵ, and this is true with probability greater than or equal to $1 - \delta$. Since decreasing δ does not result in a large increase in the number of samples, we can select δ on the order of 10^{-4} or smaller in practice.

In summary, for user-selected values of ϵ and δ, we can compute the minimum number of required samples N using (13). We then evaluate $\hat{E}_N$ in (12) by querying the one-step feasibility oracle $\phi(x^{(i)})$ for all $i = 1, \dots, N$. If $\hat{E}_N = 0$, then we can invoke the probability bound (14) to assert that $\mathcal{C}$ is *ϵ-control invariant*. Smaller values of ϵ will imply a larger N; however, this will provide a stronger guarantee on the probability that the control invariant condition is satisfied for $\mathcal{C}$. We recommend running this probabilistic control invariance test after Algorithm 1 has converged. If this test fails, one must then repeat the overall procedure by further augmenting the labeled dataset. On our tested example problems, we have found that only a small number of repeats are needed to identify a converged solution, though we have not yet provided a formal proof of convergence.

V. Case Study: Two-tank System

The proposed approach is demonstrated on a simulation case study involving two constant-volume tanks in series [28], as shown in Fig. 2. The hybrid nature of the system dynamics stems from the fact that the gas flow across each of the valves (denoted by v_i, where the subscript uniquely identifies the valves at the inlet, outlet, and between tanks 1 and 2) exhibits discrete switching between two flow regimes, namely *low pressure-drop flow* and *choke flow*. This depends on the relative magnitudes of the tank pressures (i.e., the states). In addition, the valves allow the gas to flow in only one direction, which gives an additional regime of zero flow across v_{12} whenever the downstream pressure is higher than the upstream pressure. As such, the system dynamics can be described by a model of form (1), which is summarized as

$$[x_{k+1}]_1 = \frac{P_{\text{atm}} t_{\text{s}}}{V_1} \left(F_k^{\text{in}}(x_k, u_k) - F_k^{12}(x_k, z_k) \right) + [x_k]_1,$$

$$[x_{k+1}]_2 = \frac{P_{\text{atm}} t_{\text{s}}}{V_2} \left(F_k^{12}(x_k, z_k) - F_k^{\text{out}}(x_k, z_k) \right) + [x_k]_2.$$

Here, $[x]_1 = [P]_1$, $[x]_2 = [P]_2$, and u is the position of valve v_{in}; $P_{\text{atm}} = 14.7$ psi; $t_{\text{s}} = 3$ s is the sampling time; $V_1 = 1$ m^3 and $V_2 = 3$ m^3 are the volume of tanks 1 and 2, respectively; and F^i denotes the flow across valve i, which is a function of the states x and the switching variables z as

$$F_k^{\text{in}}(x_k, u_k) = c^{\text{in}} u \left(P_{\text{sup}} - [x_k]_1 \right),$$

$$F_k^{12}(x_k, z_k) = c^{12} \sqrt{z_k^{12}},$$

$$F_k^{\text{out}}(x_k, z_k) = c^{\text{out}} \sqrt{z_k^{\text{out}}},$$

$$z_k^{12} = \begin{cases} 0.5[x_k]_1, & \text{if } [x_k]_1 > 2[x_k]_2, \\ [x_1]_k - [x_2]_k, & \text{if } [x_k]_2 \leq [x_k]_1 \leq 2[x_k]_2, \\ 0, & \text{if } [x_k]_1 \leq [x_k]_2, \end{cases}$$

$$z_k^{\text{out}} = \begin{cases} 0.5[z_k]_2, & \text{if } [x_k]_2 > 2P_{\text{atm}}, \\ [x_2]_k - P_{\text{atm}}, & \text{if } P_{\text{atm}} \leq [x_k]_2 \leq 2P_{\text{atm}}, \end{cases}$$

where $P_{\text{sup}} = 60$ psig is the pressure of the upstream gas, $c_{\text{in}} = 0.01$, $c_{12} = 0.025$, and $c_{\text{out}} = 0.025$.

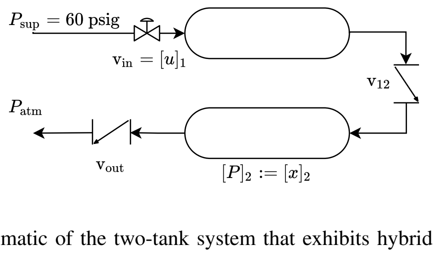

Fig. 2. Schematic of the two-tank system that exhibits hybrid dynamics.

TABLE I

Hyperparameters of the deep neural network classifier selected using grid search.

Number of Hidden Layers	4
Neurons per Layer	20
Activation Function	ReLU
Regularization Parameter α	0.0
Dropoout Rate	0.0
Learning Rate	0.001
Batch Size	32
Epochs	1000

First, we solved the feasibility problem (6) to generate the initial training data, which consist of an array of initial states x_k and the corresponding value of feasibility oracle (7). The state and control input constraints were $[10, 10]^\top \leq x_k \leq [60, 60]^\top$ and $0 \leq u_k \leq 1$, respectively. The terminal constraint $\mathbb{X}_f$ was chosen as the steady-state manifold of the dynamics. The resulting MINLP was implemented using the Pyomo optimization modeling language [29], and solved to global optimality using the Baron solver [30]. The data were then used to learn a DNN classifier, whose hyperparameters were optimized to an initial set of 20 training sample points. The DNN was trained using the Adam optimizer, and its hyperparameters are given in Table I. To assess classification performance of the DNN, we used the *precision* metric

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}.$$

Thus, the model with the best precision will be the one that maximizes true positives while minimizing false positives. Lastly, as described in Section IV, a threshold probability π_{th} was used as the decision boundary of the classifier. To mitigate false positives, the decision threshold probability was chosen to be $0.5 \leq \pi_{\text{th}} \leq 1$.[1] Note that the error tolerance ε_{tol} in Algorithm 1 does not depend on π_{th}.

Fig. 3 shows the performance of the DNN classifier trained based on 20 samples against a test dataset of 1000 samples. As expected, the performance of this classifier is inadequate and very sensitive to the random samples; its precision is quantified as 75.5% with 37 infeasible samples misclassified as feasible (i.e., false positives). To improve upon this DNN classifier, we employ the proposed AL approach (Algorithm 1) by adding 10 samples at every iteration, which are selected by maximizing the entropy function (10). It is important to note that Algorithm 1 does not require one to estimate the true precision rate, which cannot be done without a large number of test samples that would be computationally too expensive to obtain for this type of problem. The proposed stopping criterion (9) is instead based on the sensitivity of the predictions of the classifier, which can be easily estimated when only a small amount of data is available.

Fig. 4 demonstrates the performance of the classifier trained using the proposed AL approach compared to a pure random search method on a test set of 1000 samples for threshold probability values $\pi_{\text{th}} = 0.99$ (top) and $\pi_{\text{th}} = 0.5$ (bottom). Using a tolerance of $\epsilon_{\text{tol}} = 0.01$, Algorithm 1 converges after 180 training samples have been selected. To ensure a fair comparison, we run the random search method under the same DNN hyperparameter settings and the same number of 180 samples (just selected randomly in the state space). An important difference between the DNNs trained using AL versus random samples is the sensitivity with respect to the probability threshold value. In particular, the control invariant set predicted by the AL-based DNN is much more confident in its predictions (see Table II). Additionally, the AL-based DNN yields significantly fewer false positives and achieves a precision value of 98.9% compared to only 89.2% for the DNN trained on randomly-selected samples.

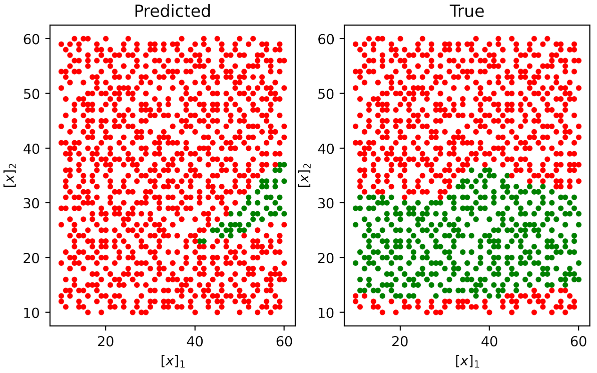

Fig. 3. Performance of the DNN classifier trained using 20 samples against a test dataset of 1000 samples. Left subplot shows the predicted decision boundaries by the classifier, whereas right subplot shows the true decision boundaries. Green and red points represent the feasible and infeasible samples, respectively.

Lastly, we employ the approach presented in Section IV to verify if the approximated set is ε-control invariant. For $\varepsilon = 0.05$ and $\delta = 0.05$, 59 random samples are required. The test was passed for all 59 randomly-selected points, implying the approximated set is invariant with 95% confidence. We verified this claim by drawing a large number of random samples that were predicted to be feasible by the DNN classifier and running the invariance test on each of these samples. Less than 1% of these samples failed the test, which is consistent with the outcome of the verification method that established less than 5% of samples should fail the test.

VI. Conclusion

We presented a sample-efficient, deep learning-based approach for data-driven estimation of invariant sets with probabilistic guarantees for invariance. The key benefit of the presented approach arises from its applicability to general

TABLE II

Sensitivity of number of feasible points predicted by DNN classifiers versus threshold value π_{th} on test data.

	$\pi_{\text{th}} = 0.5$	$\pi_{\text{th}} = 0.99$
Active learning	395	348
Random sampling	456	105

[1] The choice of π_{th} depends on the problem at hand and, thus, can be treated as a tuning parameter of the classifier.

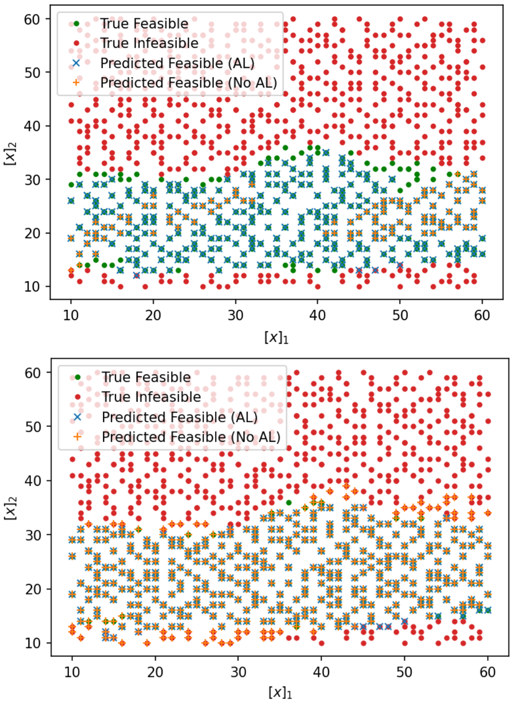

Fig. 4. Performance of the DNN classifiers trained using active learning (AL) and randomly-selected samples against a test dataset of 1000 samples for the decision threshold probability values $\pi_{\text{th}} = 0.99$ (top) and $\pi_{\text{th}} = 0.5$ (bottom). Performance is demonstrated in terms of feasible points predicted by the classifiers. Both classifiers were trained using the same hyperparameters and number of samples.

constrained nonlinear systems, irrespective of the structure, complexity, and dimension of system dynamics. Our results illustrated the importance of active learning for data-efficient learning of the feasibility oracle, in particular when performing the feasibility test hinges on solving (expensive) mixed-integer nonlinear programs. Our future work will focus on establishing rigorous convergence guarantees for the proposed approach, as well as improved tests for control invariance of approximated invariant set representations.

REFERENCES

[1] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, Switzerland, 2008.

[2] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 176–188, 2021.

[3] A. D. Bonzanini, J. A. Paulson, and A. Mesbah, "Safe learning-based model predictive control under state-and input-dependent uncertainty using scenario trees," in *Proceedings of the 59th IEEE Conference on Decision and Control*, Jeju Island, Republic of Korea, 2020, pp. 2448–2454.

[4] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2021.

[5] C. E. T. Dorea and J. Hennet, "(A, B)-invariant polyhedral sets of linear discrete-time systems," *Journal of Optimization Theory and Applications*, vol. 103, no. 3, pp. 521–542, 1999.

[6] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[7] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge, 2001.

[8] S. Rakovic, P. Grieder, M. Kvasnica, D. Mayne, and M. Morari, "Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004, pp. 1418–1423.

[9] J. M. Bravo, D. Limón, T. Alamo, and E. F. Camacho, "On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach," *Automatica*, vol. 41, no. 9, pp. 1583–1589, 2005.

[10] J. M. Bravo, T. Alamo, and E. F. Camacho, "Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets," *Automatica*, vol. 42, no. 10, pp. 1745–1751, 2006.

[11] L. Berardi, E. D. Santis, and M. D. D. Benedetto, "Invariant sets and control synthesis for switching systems with safety specifications," in *Proceedings of the International Workshop on Hybrid Systems: Computation and Control*. Springer, 2000, pp. 59–72.

[12] W. Esterhuizen, T. Aschenbruck, and S. Streif, "On maximal robust positively invariant sets in constrained nonlinear systems," *Automatica*, vol. 119, p. 109044, 2020.

[13] T. Gurriet, M. Mote, A. Singletary, E. Feron, and A. D. Ames, "A scalable controlled set invariance framework with practical safety guarantees," in *Proceedings of the IEEE Conference on Decision and Control*, Nice, France, 2019, pp. 2046–2053.

[14] A. Chakrabarty, A. Raghunathan, S. Di Cairano, and C. Danielson, "Data-driven estimation of backward reachable and invariant sets for unmodeled systems via active learning," in *Proceedings of the IEEE Conference on Decision and Control*, Miami, 2018, pp. 372–377.

[15] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Xi'an, China, 2021, pp. 1817–1824.

[16] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proceedings of the IEEE Conference on Decision and Control*, Cancun, 2008, pp. 4042–4048.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, Cambridge, 2016.

[18] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized algorithms for analysis and control of uncertain systems: With applications*. London, UK: Springer Science & Business Media, 2012.

[19] D. Q. Mayne and W. Schroeder, "Robust time-optimal control of constrained linear systems," *Automatica*, vol. 33, no. 12, pp. 2103–2118, 1997.

[20] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[21] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[22] B. Settles, "Active learning literature survey," 2009.

[23] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–40, 2021.

[24] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*. Springer, 1994, pp. 3–12.

[25] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden Markov models for information extraction," in *Proceedings of the International Symposium on Intelligent Data Analysis*, 2001, pp. 309–318.

[26] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[27] R. Tempo, E.-W. Bai, and F. Dabbene, "Probabilistic robustness analysis: Explicit bounds for the minimum number of samples," in *Proceedings of the IEEE Conference on Decision and Control*, 1996, pp. 3424–3428.

[28] C. E. Long, P. K. Polisetty, and E. P. Gatzke, "Deterministic global optimization for nonlinear model predictive control of hybrid dynamic systems," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 13, pp. 1232–1250, 2007.

[29] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff, *Pyomo–Optimization modeling in python*. Springer Science & Business Media, 2021.

[30] N. V. Sahinidis, *BARON 21.1.13: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017.