

# Learning Robust Output Control Barrier Functions from Safe Expert Demonstrations\*

Lars Lindemann<sup>†1</sup>, Alexander Robey<sup>†1</sup>, Lejun Jiang<sup>1</sup>, Stephen Tu<sup>2</sup>, and Nikolai Matni<sup>1</sup>

<sup>1</sup>Department of Electrical and Systems Engineering, University of Pennsylvania

<sup>2</sup>Google Brain Robotics

November 22, 2021

## Abstract

This paper addresses learning safe control laws from expert demonstrations. We assume that appropriate models of the system dynamics and the output measurement map are available, along with corresponding error bounds. We first propose robust output control barrier functions (ROCBFs) as a means to guarantee safety, as defined through controlled forward invariance of a safe set. We then present an optimization problem to learn ROCBFs from expert demonstrations that exhibit safe system behavior, e.g., data collected from a human operator. Along with the optimization problem, we provide verifiable conditions that guarantee validity of the obtained ROCBF. These conditions are stated in terms of the density of the data and on Lipschitz and boundedness constants of the learned function and the models of the system dynamics and the output measurement map. When the parametrization of the ROCBF is linear, then, under mild assumptions, the optimization problem is convex. We validate our findings in the autonomous driving simulator CARLA and show how to learn safe control laws from RGB camera images.

## 1 Introduction

Robust control can account for uncertainties in the system dynamics and the output measurement map, which is especially important in safety-critical applications. In particular, system and measurement map models can be obtained from first principles, or estimated from data, along with uncertainty sets that describe model error bounds. Importantly, note that it is often easy to get access to data that exhibit safe system behavior for safety-critical systems. For example, consider the scenario in which an autonomous car equipped with noisy sensors navigates through urban traffic [1]. The exact state of the car and the environment is not known but can be estimated, e.g., from a dashboard camera. Furthermore, the dynamics of the car are not perfectly known, e.g., friction coefficients may vary, but can be accurately modeled. Note also the availability of safe expert data as almost every car manufacturer records their test data. As another motivating example, consider a group of autonomous robots navigating in unknown environments [2]. Motivated

---

\*This work is funded in part by NSF awards CPS-2038873 and CAREER award ECCS-2045834, and a Google Research Scholar award.

<sup>†</sup>Lars Lindemann and Alexander Robey contributed equally.

by these observations, we propose a data-driven and model-based approach to learning safe control laws. Particularly, safety is here defined as the ability of a system to stay within a safe set. The safe set consists of the set of safe states, e.g., states that satisfy a minimum safety distance.

**Contributions:** We address the problem of finding safe output feedback control laws for partially unknown nonlinear systems from safe expert demonstrations. First, we present the notion of robust output control barrier functions as a means to establish safety in the presence of system dynamic and measurement map uncertainties. In the definition of ROCBFs, we use models of the unknown system dynamics and the unknown measurement map along with error bounds that quantify their permissible errors. Second, we propose an optimization-based approach for constructing ROCBFs from safe expert demonstrations. We further provide verifiable conditions that guarantee the validity of the obtained ROCBF. Third, we present an empirical validation of our proposed approach in the autonomous driving simulator CARLA [3].

**Paper structure:** We discuss related work in Section 2 and present background and problem formulation in Section 3. Section 4 then introduces ROCBFs, while Section 5 presents our main result on how ROCBFs can be learned from expert demonstrations. We discuss practical aspects of solving the constrained optimization problem in Section 6 and present the simulation results in Section 7. We conclude and summarize in Section 8.

## 2 Related Work

Control barrier functions (CBFs) were introduced in [4] to render a safe set controlled forward invariant. The notion of CBFs was refined by introducing reciprocal [5] and zeroing CBFs [6], which do not require that sub-level sets of the CBF be invariant within the safe set. A CBF defines a set of safe control inputs, and in particular can be used to find a minimally invasive safety-preserving correction to a nominal control law. Such a correction can be found by solving a convex quadratic program for input affine control systems. Many variations and extensions of CBFs appeared in the literature, e.g., composition of CBFs [7], CBFs for multi-robot systems [8], CBFs encoding temporal logic constraints [9], and CBFs for systems with higher relative degree [10].

More in the spirit of this paper, CBFs that account for uncertainties in the system dynamics were considered in two ways. Works [11] and [12] consider input-to-state safety to quantify possible safety violation in terms of the size of the uncertainty. Conversely, [13] proposes robust CBFs to guarantee robust safety by accounting for all permissible errors within an uncertainty set and with respect to a given model. While such an approach is in general conservative, conservatism can be reduced by better models. CBFs that account for measurement map uncertainties were first proposed in [14]. Relying on the same notion of measurement robust CBFs as in [14], the authors in [15] present empirical evaluations on a segway. Similar to the notion of ROCBF that we present in this paper, the authors in [16] consider additive disturbances in the system dynamics and state-estimation errors jointly. While the focus in [16] is, however, on achieving multiple tasks under input constraints and finite time convergence requirements, we consider more general forms of uncertainties in the system dynamics and the measurement map.

**Learning with CBFs:** CBFs were used in more data-driven and machine learning oriented contexts. These works typically assume that a CBF is already given, while we in this paper focus on constructing CBFs so that our approach should be viewed as complementary. In [17], it is shown how safe and optimal reward functions can be obtained and how these are related to CBFs. The authors in [18] learn a provably correct neural network safety guard for kinematic bicycle

**models using CBFs as safety filters.** The authors in [19] consider that the uncertainty enters the system dynamics linearly and propose to use robust adaptive CBFs, as originally presented in [20], in conjunction with online set membership identification methods. The authors in [21] consider systems perturbed by additive and multiplicative noise modeled as convex sets, and show how these sets can be estimated online using Gaussian process regression. The authors in [22] collect data to update the system model and consequently update the CBF controller episodically. A similar idea is followed in [23] where the notion of projection-to-state safety is used and where instead a projection (with respect to the CBF condition) is episodically learned. Imitation learning under safety constraints imposed by a Lyapunov function was proposed in [24]. Further work in this direction can be found in [25–27].

**Learning for CBFs:** While CBFs provide strong guarantees in terms of the safety of the system, an open problem that has not been fully addressed in prior work is how such CBFs can be constructed for general classes of systems. Indeed, the lack of principled and systematic methods to constructs CBFs is one of the main bottlenecks in applying CBFs. This challenge is similarly present when addressing stability using control Lyapunov functions [28]. For certain types of mechanical systems under input constraints, analytic CBFs can be constructed [29]. The construction of polynomial barrier functions towards certifying safety for polynomial systems by using sum-of-squares (SOS) programming was proposed in [30]. Finding CBFs poses additional challenges in terms of the control input resulting in bilinear SOS programming as presented in [31, 32] and summarized in [33]. The work in [34] considers the construction of higher order CBFs and their composition by, similarly to [31, 32], alternating-descent heuristics to solve the arising bilinear SOS program. Such SOS-based approaches, however, are known to be limited in scalability and do not use available data.

A promising research direction, in the spirit of what we propose in this paper, is to learn CBFs from data. The authors in [35] construct CBFs from safe and unsafe data using support vector machines, while [36] learns a set of linear CBFs for clustered datasets. Recently, the authors in [37] proposed learning limited duration CBFs in order to achieve limited-duration safety. In [38], a neural network controller is trained episodically to imitate an already given CBF. The authors in [39] learn parameters associated with the constraints of a CBF to improve feasibility. These works present empirical validations, while no formal correctness guarantees are provided. In [40], motion primitives are learned from expert demonstrations that are stabilized using control Lyapunov functions. The authors in [41–43] propose counter-example guided approaches to learn Lyapunov and barrier functions for known closed-loop systems, while [44] learn Lyapunov functions from data and without system knowledge. In [45], a control law is learned jointly with a barrier function, and safety is verified post-hoc using Lipschitz arguments. A similar approach of learning a safe controller jointly with a barrier function followed by post-hoc validation using satisfiability modulo theory solver is presented in [46]. As opposed to these works, we here explicitly make use of safe expert demonstrations. Expert trajectories are utilized in [47] to learn a contraction metric, ensuring incremental stability, along with a tracking controller by providing correctness guarantees based on Lipschitz arguments. Towards incorporating uncertainty in the environment, the work in [48] learns signed distance fields that are then used to define the CBF under consideration of potential estimation errors. A learning framework for constructing decentralized CBFs for multi-agent systems appeared in [49]. In our previous work [50], we proposed to learn CBFs for known nonlinear systems from expert demonstrations. We provided the first conditions that ensure correctness of the learned CBF using Lipschitz continuity and covering number arguments. In [51] and [52], we

extended this framework to partially unknown hybrid systems. In this paper, we instead focus on unknown output measurement maps and output feedback control laws for partially unknown nonlinear systems.

### 3 Background and Problem Formulation

Let  $\mathbb{R}$  and  $\mathbb{R}_{\geq 0}$  be the set of real and non-negative real numbers, respectively, and  $\mathbb{R}^n$  the set of  $n$ -dimensional real vectors. The inner-product between two vectors  $x, y \in \mathbb{R}^n$  is denoted by  $\langle x, y \rangle$ . Let  $\|\cdot\|$  be a norm and let  $\|\cdot\|_*$  denote its dual norm while  $\|\cdot\|$  denotes its induced matrix norm. For  $\epsilon > 0$ , we let  $\mathcal{B}_\epsilon(\bar{x}) := \{x \in \mathbb{R}^n \mid \|x - \bar{x}\| \leq \epsilon\}$  be a closed norm ball around a point  $\bar{x} \in \mathbb{R}^n$ . For a given set  $\mathcal{C}$ , we denote by  $\text{bd}(\mathcal{C})$  and  $\text{int}(\mathcal{C})$  the boundary and interior of  $\mathcal{C}$ , respectively. For two sets  $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{R}^n$ , let their Minkowski sum be  $\mathcal{C}_1 \oplus \mathcal{C}_2 := \{x_1 + x_2 \in \mathbb{R}^n \mid x_1 \in \mathcal{C}_1, x_2 \in \mathcal{C}_2\}$ . A continuous function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  is an extended class  $\mathcal{K}$  function if it is strictly increasing with  $\alpha(0) = 0$ . For a function  $F : A \rightarrow B$  and sets  $\mathcal{A} \subseteq A$  and  $\mathcal{B} \subseteq B$ , let  $F(\mathcal{A}) := \{b \in B \mid \exists a \in \mathcal{A}, F(a) = b\}$  denote the image of  $\mathcal{A}$  under  $F$ . We adopt the convention to denote the Lipschitz constant of a multivariable function  $F : A \times B \rightarrow C$  for a fixed  $a \in A$  and within the set  $\mathcal{B} \subseteq B$  as  $\text{Lip}_F(a, \mathcal{B})$ . We remark that the proofs of our technical lemmas, propositions, and theorems can be found in the appendices.

#### 3.1 System Description

At time  $t \in \mathbb{R}_{\geq 0}$ , let  $x(t) \in \mathbb{R}^n$  be the state of the dynamical control system that is described by the set of equations

$$\dot{x}(t) = f(t) + g(t)u(t), \quad (1a)$$

$$x(0) := x_0 \quad (\text{initial condition}), \quad (1b)$$

$$f(t) := F(x(t), t) \quad (\text{internal dynamics}), \quad (1c)$$

$$g(t) := G(x(t), t) \quad (\text{input dynamics}), \quad (1d)$$

$$y(t) := Y(x(t)) \quad (\text{output measurements}), \quad (1e)$$

$$u(t) := U(y(t), t) \quad (\text{output feedback control law}), \quad (1f)$$

where  $x_0 \in \mathbb{R}^n$  is an initial condition of the system. The functions  $F : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  and  $G : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times m}$  are in general *only partially known* and locally Lipschitz continuous in the first and piecewise continuous and bounded in the second argument. The reason for the functions  $F$  and  $G$  only been partially known may be due to unmodeled internal or input dynamics, modeling errors, or noise.

**Assumption 1.** We assume to have models  $\hat{F} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  and  $\hat{G} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times m}$  of  $F(x, t)$  and  $G(x, t)$  together with functions  $\Delta_F : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  and  $\Delta_G : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  that bound their respective errors as

$$\begin{aligned} \|\hat{F}(x, t) - F(x, t)\| &\leq \Delta_F(x, t) \text{ for all } (x, t) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0}, \\ \|\hat{G}(x, t) - G(x, t)\| &\leq \Delta_G(x, t) \text{ for all } (x, t) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0}. \end{aligned}$$

The functions  $\hat{F}(x, t)$ ,  $\hat{G}(x, t)$ ,  $\Delta_F(x, t)$  and  $\Delta_G(x, t)$  are assumed to be locally Lipschitz continuous in the first and piecewise continuous and bounded in the second argument.

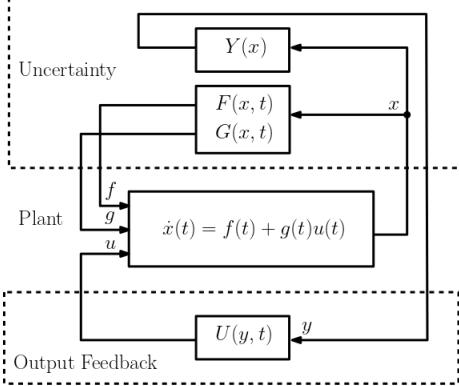


Figure 1: System Overview: Output feedback loop (bottom) and uncertainty loop (top) affecting the plant (middle). The internal and input dynamics  $F$  and  $G$  are constrained as per Assumption 1, while the output measurement map  $Y$  is constrained as per Assumption 2.

Such models  $\hat{F}(x, t)$  and  $\hat{G}(x, t)$  together with error bounds  $\Delta_F(x, t)$  and  $\Delta_G(x, t)$  may be obtained by identifying model parameters or by system identification techniques. Let us now define the set of *admissible system dynamics* according to Assumption 1 as

$$\begin{aligned}\mathcal{F}(x, t) &:= \{f \in \mathbb{R}^n \mid \|f - \hat{F}(x, t)\| \leq \Delta_F(x, t)\}, \\ \mathcal{G}(x, t) &:= \{g \in \mathbb{R}^{n \times m} \mid \|g - \hat{G}(x, t)\| \leq \Delta_G(x, t)\}.\end{aligned}$$

The output measurement map  $Y : \mathbb{R}^n \rightarrow \mathbb{R}^p$  is *only partially known*, but assumed to be locally Lipschitz continuous. We assume that there exists an inverse map  $X : \mathbb{R}^p \rightarrow \mathbb{R}^n$  that recovers the state  $x$  as  $X(Y(x)) = x$ . This assumption means that a measurement  $y$  uniquely defines a corresponding state  $x$  and implies that  $p \geq n$ . This way, we implicitly assume high-dimensional measurements  $y$ . For instance, these can include camera images where the inverse map  $X$  recovers the position of the system or even a sequence of camera images to recover its velocity. We present an example in the simulation study and refer to related literature using similar assumptions, such as [14]. If the function  $Y$  was known, which is not the case here, one could construct  $X$  to recover the state  $x$  from an output measurements  $y$ . However, oftentimes the output measurement map  $Y$  and hence the inverse map  $X$  are not known.

**Assumption 2.** We assume to have a model  $\hat{X} : \mathbb{R}^p \rightarrow \mathbb{R}^n$  that estimates  $X(y)$  together with a function  $\Delta_X : \mathbb{R}^p \rightarrow \mathbb{R}_{\geq 0}$  that bounds the error between  $\hat{X}(y)$  and  $X(y)$  as

$$\|\hat{X}(y) - X(y)\| \leq \Delta_X(y) \text{ for all } y \in Y(\mathbb{R}^n).$$

The functions  $\hat{X}(y)$  and  $\Delta_X(y)$  are assumed to be locally Lipschitz continuous.

The error bound  $\Delta_X(y)$  can, as shown in [14] under the assumption that  $\mathcal{U} = \mathbb{R}^m$ , be quantified from finite data with high probability. Let us now define the set of *admissible inverse output measurement maps* according to Assumption 2 as

$$\mathcal{X}(y) := \{x \in \mathbb{R}^n \mid \|\hat{X}(y) - x\| \leq \Delta_X(y)\}.$$

Symbol	Meaning
$F(x, t), G(x, t)$	Underlying true yet unknown internal and input dynamics.
$\mathcal{F}(x, t), \mathcal{G}(x, t)$	Admissible set of internal and input dynamics.
$\hat{F}(x, t), \hat{G}(x, t)$	Model of internal and input dynamics.
$\Delta_F(x, t), \Delta_G(x, t)$	Error bounds quantifying mismatch between the true internal and input dynamics and the models.
$U(y, t)$	Output feedback control law.
$Y(x)$	Underlying true yet unknown output measurement map.
$X(y)$	Unknown inverse output measurement map $Y$ so that $X(Y(x)) = x$ .
$\mathcal{X}(y)$	Admissible set of inverse output measurement maps.
$\hat{X}(y)$	Model of the inverse output measurement map $X(y)$ .
$\Delta_X(y)$	Error bound quantifying mismatch between the inverse output measurement map and its inverse.
$h(x)$ and $\mathcal{C}$	The ROCBF $h(x)$ defines the safe set $\mathcal{C}$ via its zero-superlevel sets.
$B(x, t, u), B_j(x, t)$ for $j \in \{1, 2, 3\}$ , $C(x, t, u)$	Functions used to define the ROCBF notion in Definition 2.
$\mathcal{S}$	The geometric safe set, e.g., defined by a minimum safety distance.
$\mathcal{D}, \mathcal{Y}$	These sets define the set of admissible states and output measurements, respectively.
$Z_{\text{dyn}}$	Expert demonstrations containing safe input-output data pairs.
$Z_{\text{safe}}, Z_{\mathcal{N}}$	Sets of safe and as unsafe labeled datapoints.
$\mathcal{N}$	Set of as unsafe labeled states.
$q(u, y, t)$	Encodes the ROCBF condition from Definition 2 used in the optimization problem (7).

Table 1: Summary of the most important notation used throughout the paper.

Finally, the function  $U : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{U} \subseteq \mathbb{R}^m$  defines the output feedback control law that is continuous in the first and piecewise continuous in the second argument. The set  $\mathcal{U}$  imposes input constraints.

The system under consideration is illustrated in Fig. 1. Let a solution to (1) under an output feedback control law  $U(y, t)$  be  $x : \mathcal{I} \rightarrow \mathbb{R}^n$  where  $\mathcal{I} \subseteq \mathbb{R}_{\geq 0}$  is the maximum definition interval of  $x$ . Solutions to (1) are guaranteed to exist under the stated assumptions on the functions  $F(x, t)$ ,  $G(x, t)$ ,  $Y(x)$ , and  $U(y, t)$ , even though uniqueness of solutions may not be guaranteed. We do not assume forward completeness of solutions to (1) under an output feedback control law  $U(y, t)$ , i.e.,  $\mathcal{I}$  may be bounded. We will, however, guarantee forward completeness with our proposed approach.

We remark that all important symbols that have been or will be introduced in this paper are summarized in Table 1.

### 3.2 Problem Formulation

The goal in this paper is to learn an output feedback control law  $U(y, t)$  such that prescribed safety properties with respect to the *geometric safe set*  $\mathcal{S} \subseteq \mathbb{R}^n$  are met by the system in (1). By geometric safe set, we mean that  $\mathcal{S}$  describes the set of safe states as naturally specified on a subset of the system configuration space (e.g., to avoid collision, vehicles must maintain a minimum separating distance). Let us first formally define what we mean by safety.

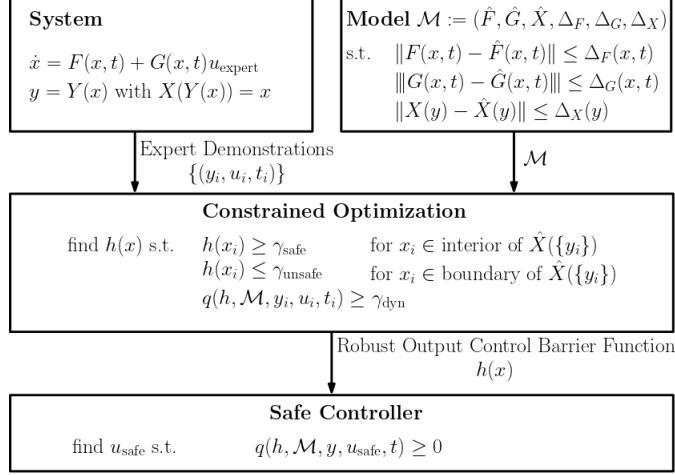


Figure 2: Framework to learn safe control laws where the functions  $F$ ,  $G$ ,  $Y$ , and  $X$  are not known. We solve a constrained optimization problem that uses expert demonstrations and a model of the unknown system.

**Definition 1.** A set  $\mathcal{C} \subseteq \mathbb{R}^n$  is said to be robustly output controlled forward invariant with respect to the system in (1) if there exist an output feedback control law  $U(y, t)$  such that, for all initial conditions  $x(0) \in \mathcal{C}$ , for all admissible system dynamics  $F(x, t) \in \mathcal{F}(x, t)$  and  $G(x, t) \in \mathcal{G}(x, t)$ , and for all admissible inverse output measurement maps  $X(y) \in \mathcal{X}(y)$ , every solution  $x(t)$  to (1) under  $U(y, t)$  is such that:

1.  $x(t) \in \mathcal{C}$  for all  $(t) \in \mathcal{I}$ , and
2. the interval  $\mathcal{I}$  is unbounded, i.e.,  $\mathcal{I} = [0, \infty)$ .

If the set  $\mathcal{C}$  is additionally contained within the geometric safe set  $\mathcal{S}$ , i.e.,  $\mathcal{C} \subseteq \mathcal{S}$ , the system in (1) is said to be safe under the safe control law  $U(y, t)$ .

Towards the goal of learning a safe control law  $U(y, t)$ , we assume to be given a set of so called expert demonstrations consisting of  $N_1$  input-output data pairs  $(u_i, y_i) \in \mathbb{R}^m \times \mathbb{R}^p$  along with a time stamp  $t_i \in \mathbb{R}_{\geq 0}$  as

$$Z_{\text{dyn}} := \{(u_i, y_i, t_i)\}_{i=1}^{N_1}$$

that were recorded when the system was in a safe state  $X(y_i) \in \text{int}(\mathcal{S})$ .

**Problem 1.** Let the system in (1) and the set of safe expert demonstrations  $Z_{\text{dyn}}$  be given. Under Assumptions 1 and 2, learn a function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  from  $Z_{\text{dyn}}$  so that the set

$$\mathcal{C} := \{x \in \mathbb{R}^n \mid h(x) \geq 0\} \tag{2}$$

is robustly output controlled forward invariant with respect to (1) and such that  $\mathcal{C} \subseteq \mathcal{S}$ , i.e., so that the system in (1) is safe.

An overview of the proposed framework to solve Problem 1 is shown in Fig. 2. We first define robust output control barrier functions (ROCBF) in Section 4, and show that a RCBF  $h(x)$  defines an output feedback control law  $U(y, t)$  that renders the set  $\mathcal{C}$  robustly controlled forward invariant with respect to the system in (1). In Section 5, we formulate a constrained optimization problem to learn a RCBF  $h(x)$  so that the learned safe set  $\mathcal{C}$  is contained within the geometric safe set  $\mathcal{S}$ , i.e.,  $\mathcal{C} \subseteq \mathcal{S}$ . The optimization problem takes the system model  $\mathcal{M} := (\hat{F}, \hat{G}, \hat{X}, \Delta_F, \Delta_G, \Delta_X)$  and the expert demonstrations  $\{(y_i, u_i, t_i)\}$  as inputs and contains several constraints on  $h(x)$  that will be derived in the sequel. Ultimately, the optimization problem produces a safe output feedback control law  $U(y, t)$ .

## 4 Robust Output Control Barrier Functions

In this section, we assume to be given a twice continuously differentiable function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$ . The goal is now to define conditions that, if satisfied, allow to construct a control law  $U(y, t)$  that renders the set  $\mathcal{C}$  robustly output controlled forward invariant with respect to the system in (1).

We assume that the set  $\mathcal{C}$  has non-empty interior, and we let  $\mathcal{Y} \subseteq \mathbb{R}^p$  be an open set such that  $\mathcal{Y} \supseteq Y(\mathcal{C})$ , or equivalently  $X(\mathcal{Y}) \supseteq \mathcal{C}$ .<sup>1</sup> The set  $X(\mathcal{Y})$  is typically the domain of interest in existing state-based CBF frameworks, see e.g., [6]. The set  $\mathcal{Y}$  is here used instead since we consider output measurements while having no direct access to the state of the system.

Towards rendering the set  $\mathcal{C}$  robustly output controlled forward invariant, we can formulate the following sufficient condition: there exists a locally Lipschitz continuous extended class  $\mathcal{K}$  function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  such that

$$\sup_{u \in \mathcal{U}} \inf_{x \in \mathcal{X}(y)} \underbrace{\inf_{\substack{f \in \mathcal{F}(x, t) \\ g \in \mathcal{G}(x, t)}} \langle \nabla h(x), f + gu \rangle}_{\text{Change in } h \text{ along true dynamics}} + \alpha(h(x)) \geq 0 \quad (3)$$

for all  $(y, t) \in \mathcal{Y} \times \mathbb{R}_{\geq 0}$ . For an observed output measurement  $y \in \mathcal{Y}$ , note in particular that the corresponding state satisfies  $X(y) \in \mathcal{X}(y)$  and that the dynamics satisfy  $F(x, t) \in \mathcal{F}(x, t)$  and  $G(x, t) \in \mathcal{G}(x, t)$  in accordance with Assumptions 1 and 2. Observe, however, that the condition in (3) is difficult to evaluate due to the existence of the infimum operators.

Towards a more tractable condition, let us first define the function  $B : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \times \mathcal{U} \rightarrow \mathbb{R}$  as

$$B(x, t, u) := \underbrace{\langle \nabla h(x), \hat{F}(x, t) + \hat{G}(x, t)u \rangle}_{\text{Change in } h \text{ along model dynamics}} + \alpha(h(x)) - \underbrace{\|\nabla h(x)\|_* (\Delta_F(x, t) + \Delta_G(x, t)\|u\|)}_{\text{Robustness term accounting for system model error}}$$

The constraint (3) can now instead be written as

$$\sup_{u \in \mathcal{U}} \inf_{x \in \mathcal{X}(y)} B(x, t, u) \geq 0$$

where the infimum operator over the set of admissible system dynamics  $\mathcal{F}(x, t)$  and  $\mathcal{G}(x, t)$  in (3) has been removed. To remove the remaining infimum operator that accounts for measurement uncertainty, let us write  $B(x, t, u)$  as

$$B(x, t, u) = B_1(x, t) + \langle B_2(x, t), u \rangle + B_3(x, t)\|u\|$$

---

<sup>1</sup>Note that the function  $Y$  is unknown. In Section 5, we construct the sets  $\mathcal{Y}$  and  $\mathcal{C}$  in a way so that  $\mathcal{Y} \supseteq Y(\mathcal{C})$  holds.

where the functions  $B_1 : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ ,  $B_2 : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ , and  $B_3 : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$  are defined as

$$\begin{aligned} B_1(x, t) &:= \langle \nabla h(x), \hat{F}(x, t) \rangle + \alpha(h(x)) - \|\nabla h(x)\|_* \Delta_F(x, t), \\ B_2(x, t) &:= \hat{G}(x, t)^T \nabla h(x), \\ B_3(x, t) &:= -\|\nabla h(x)\|_* \Delta_G(x, t). \end{aligned}$$

Next, denote, for a fixed time  $t \in \mathbb{R}_{\geq 0}$ , the local Lipschitz constants of the functions  $B_1(x, t)$ ,  $B_2(x, t)$ , and  $B_3(x, t)$  within the set  $\mathcal{X} \subseteq \mathbb{R}^n$  by  $\text{Lip}_{B_1}(\mathcal{X}, t)$ ,  $\text{Lip}_{B_2}(\mathcal{X}, t)$ , and  $\text{Lip}_{B_3}(\mathcal{X}, t)$ , respectively. Let  $C : \mathbb{R}^p \times \mathbb{R}_{\geq 0} \times \mathcal{U}$  be the local Lipschitz constant of  $B(x, t, u)$  expressed as

$$C(y, t, u) := \text{Lip}_{B_1}(\mathcal{X}(y), t) + \text{Lip}_{B_2}(\mathcal{X}(y), t) \|u\|_* + \text{Lip}_{B_3}(\mathcal{X}(y), t) \|u\|$$

that will be used to account for measurement uncertainty and the fact that the functions  $X$  and  $Y$  are unknown. We are now ready to present the definition of a ROCBF that will guarantee that the set  $\mathcal{C}$  is robustly output controlled forward invariant.

**Definition 2.** *The function  $h(x)$  is said to be a robust output control barrier function (ROCBF) on an open set  $\mathcal{Y} \supseteq Y(\mathcal{C})$  if there exist a locally Lipschitz continuous extended class  $\mathcal{K}$  function  $\alpha : \mathbb{R} \rightarrow \mathbb{R}^2$  such that*

$$\sup_{u \in \mathcal{U}} B(\hat{X}(y), t, u) - \underbrace{C(y, t, u) \Delta_X(y)}_{\substack{\text{Robustness term accounting} \\ \text{for output measurement error}}} \geq 0 \quad (4)$$

for all  $(y, t) \in \mathcal{Y} \times \mathbb{R}_{\geq 0}$ .

Note that the standard CBF condition from [6] is retained when  $X(y) = \hat{X}(y)$ ,  $F(x, t) = \hat{F}(x, t)$ ,  $G(x, t) = \hat{G}(x, t)$ ,  $\Delta_X(y) = 0$ ,  $\Delta_F(x, t) = 0$ , and  $\Delta_G(x, t) = 0$ . We can define the set of safe control inputs induced by a ROCBF  $h(x)$  to be

$$K_{\text{ROCBF}}(y, t) := \{u \in \mathbb{R}^m \mid B(\hat{X}(y), t, u) - C(y, t, u) \Delta_X(y) \geq 0\}.$$

The next theorem is the main result of this section and shows under which conditions an output feedback control law  $U(y, t) \in K_{\text{ROCBF}}(y, t)$  renders the set  $\mathcal{C}$  robustly output controlled forward invariant.

**Theorem 1.** *Assume that  $h(x)$  is a ROCBF on the set  $\mathcal{Y}$  that is such that  $\mathcal{Y} \supseteq Y(\mathcal{C})$ . Assume also that the function  $U : \mathcal{Y} \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{U}$  is continuous in the first and piecewise continuous in the second argument and such that  $U(y, t) \in K_{\text{ROCBF}}(y, t)$ . Then  $x(0) \in \mathcal{C}$  implies  $x(t) \in \mathcal{C}$  for all  $t \in \mathcal{I}$ . If the set  $\mathcal{C}$  is compact, it follows that  $\mathcal{C}$  is robustly output controlled forward invariant under  $U(y, t)$ , i.e.,  $\mathcal{I} = [0, \infty)$ .*

## 5 Learning ROCBFs from Expert Demonstrations

The previous section provides strong safety guarantees when  $h(x)$  is a ROCBF and when the set  $\mathcal{C}$ , which is defined through  $h(x)$ , is such that  $\mathcal{C} \subseteq \mathcal{S}$ . However, one is still left with the potentially

---

<sup>2</sup>Recall that  $\alpha$  is contained within the functions  $B(x, t, u)$  and  $B_1(x, t)$ .

difficult task of finding a twice continuously differentiable function  $h(x)$  such that (i) the set  $\mathcal{C}$  defined in equation (2) is contained within the set  $\mathcal{S}$ , while having a sufficiently large volume, and (ii) it satisfies the barrier constraint (4) on an open set  $\mathcal{Y}$  that is such that  $\mathcal{Y} \supseteq Y(\mathcal{C})$ . While safety constraints are often naturally specified on a subset of the configuration space of a system, e.g., to avoid collision, vehicles must maintain a minimum distance, ensuring that a function  $h(x)$  specified using such geometric intuition is a ROCBF, i.e., satisfying the constraint (4), can involve verifying complex relationships between the vector fields  $\hat{F}(x, t)$  and  $\hat{G}(x, t)$ , the state estimate  $\hat{X}(y)$ , the ROCBF  $h(x)$ , and its gradient  $\nabla h(x)$ , while accounting for the error bounds  $\Delta_X(y)$  as well as  $\Delta_F(x, t)$  and  $\Delta_G(x, t)$ .

This challenge motivates the approach taken in this paper, wherein we propose an optimization-based approach to learning a ROCBF from safe expert demonstrations for a system with partially known dynamics.

## 5.1 The Datasets

Let us first define the finite set of safe datapoints

$$Z_{\text{safe}} := \bigcup_{(u_i, y_i, t_i) \in Z_{\text{dyn}}} \hat{X}(y_i)$$

as the projection of all datapoints  $y_i$  in  $Z_{\text{dyn}}$  via the model of the inverse output measurement map  $\hat{X}$  into the state domain. For  $\epsilon > 0$ , define the set of admissible states  $\mathcal{D} \subseteq \mathbb{R}^n$  as

$$\begin{aligned} \mathcal{D}' &:= \bigcup_{x_i \in Z_{\text{safe}}} \mathcal{B}_\epsilon(x_i) \\ \mathcal{D} &:= \mathcal{D}' \setminus \text{bd}(\mathcal{D}'). \end{aligned}$$

The set  $\mathcal{D}'$  is the union of  $\epsilon$  norm balls wrapped around each safe datapoint  $x_i \in Z_{\text{safe}}$ , see Fig. 3 (left and centre). The set of admissible states  $\mathcal{D}$  is equivalent to the set  $\mathcal{D}'$  without its boundary so that  $\mathcal{D}$  is open by definition. We additionally assume that the set  $\mathcal{D}$  is such that  $\mathcal{D} \subseteq \mathcal{S}$ , which can be easily achieved even when datapoints  $\hat{X}(y_i)$  are close to  $\text{bd}(\mathcal{S})$  by adjusting  $\epsilon$  or by omitting  $y_i$  from  $Z_{\text{dyn}}$  in the definition of  $Z_{\text{safe}}$ . This additional requirement is necessary to later ensure safety in the sense that the learned safe set is such that  $\mathcal{C} \subseteq \mathcal{S}$ .

**Remark 1.** First, note that we define the set  $\mathcal{D}$  based on expert demonstrations  $y_i$ , via the transformation  $\hat{X}(y_i)$  into the state domain, for which control inputs  $u_i$  are available so that these can later be used for learning a safe control law. Second, the by  $\hat{X}$  transformed  $y$  components of  $Z_{\text{dyn}}$  define an  $\epsilon$ -net of  $\mathcal{D}$  by definition or, equivalently, the set  $Z_{\text{safe}}$  is an  $\epsilon$ -net of  $\mathcal{D}$ . In other words, for each  $x \in \mathcal{D}$ , there exists a  $y_i$  in  $(u_i, y_i, t_i) \in Z_{\text{dyn}}$  such that  $\|\hat{X}(y_i) - x\| \leq \epsilon$ . Finally, conditions on  $\epsilon$  will be specified later to ensure the validity of the learned safe output feedback control law.

We define the set of admissible output measurements as

$$\mathcal{Y} := Y(\mathcal{D}),$$

i.e., as the projection of the set  $\mathcal{D}$  under the unknown output measurement map  $Y$ . We remark that the set  $\mathcal{Y}$ , illustrated in Fig. 3 (left), is unknown as the output measurement map  $Y$  is unknown.

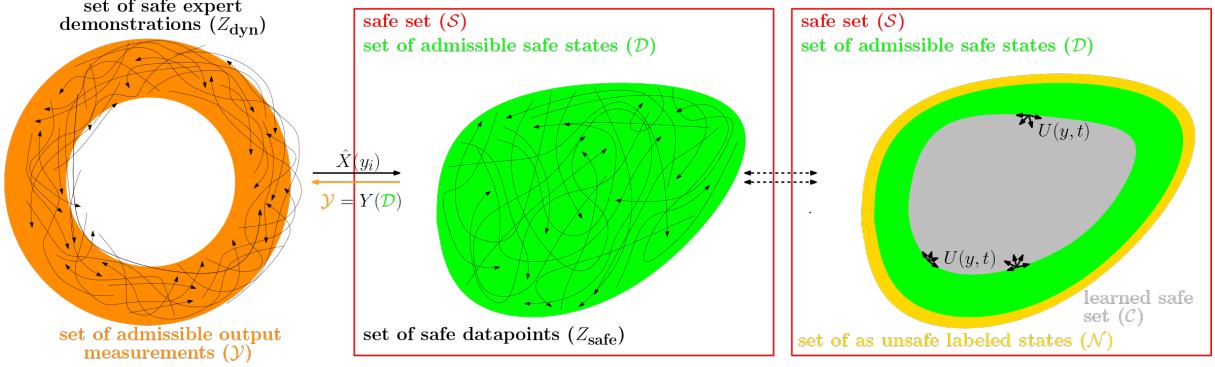


Figure 3: Problem Setup (left): The set of observed safe expert demonstrations  $Z_{\text{dyn}}$  (black lines) based on which we want to learn a safe control law. Also shown, the set of admissible output measurements  $\mathcal{Y}$  (orange ring) that is defined based on the set of admissible safe states  $\mathcal{D}$  via the unknown output measurement map  $Y$ . Transformation into state domain (centre): The geometric safe set  $\mathcal{S}$  (red box) and the set of admissible safe states  $\mathcal{D}$  (green region) that is defined as the union of  $\epsilon$ -balls that are wrapped around the via the model of the inverse output measurement map  $\hat{X}$  projected  $y$  components of  $Z_{\text{dyn}}$ . Learned safe set (right): The set of as unsafe labeled states  $\mathcal{N}$  (golden ring), defined around  $\mathcal{D}$ , ensures that the learned safe set  $\mathcal{C}$  (gray region), which is defined via the learned RBCF  $h(x)$ , is such that  $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{S}$ .

Note also that the set  $\mathcal{Y}$  is open as required in Theorem 1 since the set  $\mathcal{D}$  is open and since the function  $Y$  is locally Lipschitz continuous.

We next define the set of as unsafe labeled states

$$\mathcal{N} := \{\text{bd}(\mathcal{D}) \oplus \mathcal{B}_\sigma(0)\} \setminus \mathcal{D},$$

for  $\sigma > 0$ , which should be thought of as a layer of width  $\sigma$  surrounding  $\mathcal{D}$ , see Fig. 3 (right) for a graphical depiction. As will be made clear in the sequel, by enforcing that the value of the learned function  $h(x)$  is negative on  $\mathcal{N}$ , we ensure that the set  $\mathcal{C}$  is contained within  $\mathcal{D}$ , and hence also within  $\mathcal{S}$ . This is why we refer to  $\mathcal{N}$  as set of as unsafe labeled states. Towards ensuring that  $h(x) < 0$  for all  $x \in \mathcal{N}$ , we assume that points

$$Z_{\mathcal{N}} := \{x_i\}_{i=1}^{N_2}$$

are sampled from the set  $\mathcal{N}$  such that  $Z_{\mathcal{N}}$  forms an  $\epsilon_{\mathcal{N}}$ -net of  $\mathcal{N}$  – conditions on  $\epsilon_{\mathcal{N}}$  will be specified in the sequel. We emphasize that no associated control inputs  $u_i$  are needed for the samples  $Z_{\mathcal{N}} \subset \mathcal{N}$ , as these points are not generated by the expert, and can instead be obtained by simple computational methods such as gridding or uniform sampling.

While the definition of the set  $\mathcal{C}$  in (2) is specified over all of  $\mathbb{R}^n$ , e.g., the definition of  $\mathcal{C}$  considers all  $x \in \mathbb{R}^n$  such that  $h(x) \geq 0$ , we make a minor modification to this definition in order to restrict the domain of interest to  $\mathcal{N} \cup \mathcal{D}$  as

$$\mathcal{C} := \{x \in \mathcal{N} \cup \mathcal{D} \mid h(x) \geq 0\}. \quad (5)$$

This restriction is natural, as we are learning a function  $h(x)$  from data sampled only over  $\mathcal{N} \cup \mathcal{D}$ , and we will show that the inclusion  $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{S}$  holds.

## 5.2 The Constrained Optimization Problem

We state the constrained optimization problem for learning safe ROCBFs upfront before we provide conditions in Section 5.3 under which a feasible solution to the optimization problem is a ROCBF.

Let  $\mathcal{H}$  be a normed function space of twice continuously differentiable functions  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  for which the local Lipschitz constant  $\text{Lip}_h(\mathcal{X})$  within a set  $\mathcal{X} \subseteq \mathbb{R}^n$  can be calculated. Let us next define the function

$$q(u, y, t) := B(\hat{X}(y), t, u) - (\bar{L}_1 + \bar{L}_2 \|u\|_\star + \bar{L}_3 \|u\|) \Delta_X(y) \quad (6)$$

according to (4). Here, however, we include positive constants  $\bar{L}_1, \bar{L}_2, \bar{L}_3 > 0$  instead of the Lipschitz constants of the functions  $B_1, B_2$ , and  $B_3$ . We will later require that these constants are upper bounds of the Lipschitz constants of  $B_1, B_2$ , and  $B_3$ , respectively. The reason for including  $\bar{L}_1, \bar{L}_2$ , and  $\bar{L}_3$  is simply that calculating the Lipschitz constants of  $B_1, B_2$ , and  $B_3$  is not possible beforehand as they depend on the function  $h$  that we try to synthesize. Note that the function  $\alpha$  is contained within  $B$  as a hyperparameter.

We propose the following constrained optimization problem to learn a ROCBF from expert demonstrations:

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}} \|h\| \quad (7a)$$

$$\text{s.t. } h(x_i) \geq \gamma_{\text{safe}}, \quad \forall x_i \in \underline{Z}_{\text{safe}} \quad (7b)$$

$$h(x_i) \leq -\gamma_{\text{unsafe}}, \quad \forall x_i \in Z_{\mathcal{N}} \quad (7c)$$

$$q(u_i, y_i, t_i) \geq \gamma_{\text{dyn}}, \quad \forall (u_i, y_i, t_i) \in Z_{\text{dyn}} \quad (7d)$$

where the set  $\underline{Z}_{\text{safe}}$  is a subset of  $Z_{\text{safe}}$ , i.e.,  $\underline{Z}_{\text{safe}} \subseteq Z_{\text{safe}}$ , as detailed in the next section and where the positive constants  $\gamma_{\text{safe}}, \gamma_{\text{unsafe}}, \gamma_{\text{dyn}} > 0$  are hyperparameters.<sup>3</sup>

## 5.3 Conditions guaranteeing learned safe ROCBFs

We now derive conditions on  $\gamma_{\text{safe}}, \gamma_{\text{unsafe}}, \gamma_{\text{dyn}}, \bar{L}_1, \bar{L}_2, \bar{L}_3, \epsilon, \epsilon_{\mathcal{N}}$ , and the local Lipschitz constants of the functions  $h, q, B_1, B_2$ , and  $B_3$  under which a feasible solution to the constrained optimization problem (7) is a safe ROCBF on  $\mathcal{Y}$ .

### 5.3.1 Guaranteeing $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{S}$

We begin with establishing the requirement that  $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{S}$ . First note that constraint (7b) ensures that the set  $\mathcal{C}$ , as defined in equation (5), has non-empty interior when  $\underline{Z}_{\text{safe}} \neq \emptyset$ . Let us next state conditions under which the constraint (7c) ensures that the learned function  $h$  from (7) satisfies  $h(x) < 0$  for all  $x \in \mathcal{N}$ , which in turn ensures that  $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{S}$ .

**Proposition 1.** *Let  $h(x)$  be Lipschitz continuous with local Lipschitz constant  $\text{Lip}_h(\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_i))$  within the set  $\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_i)$  for  $x_i \in Z_{\mathcal{N}}$ . Let  $\gamma_{\text{unsafe}} > 0$ ,  $Z_{\mathcal{N}}$  be an  $\epsilon_{\mathcal{N}}$ -net of  $\mathcal{N}$ , and let*

$$\epsilon_{\mathcal{N}} < \frac{\gamma_{\text{unsafe}}}{\text{Lip}_h(\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_i))} \quad (8)$$

for all  $x_i \in Z_{\mathcal{N}}$ . Then, the constraint (7c) ensures that  $h(x) < 0$  for all  $x \in \mathcal{N}$ .

---

<sup>3</sup>We remark that instead of a global constant  $\gamma_{\text{safe}}$ , we could choose  $\gamma_{\text{safe}}$  locally for each datapoint, i.e., using the set of hyperparameters  $\{\gamma_{\text{safe},i}\}_{i=1}^{N_1}$ . The same applies to the hyperparameters  $\gamma_{\text{unsafe}}$  and  $\gamma_{\text{dyn}}$ .

In summary, Proposition 1 says that a larger Lipschitz constant of the function  $h$  requires a larger margin  $\gamma_{\text{unsafe}}$  and/or a finer net of unsafe datapoints as indicated by  $\epsilon_{\mathcal{N}}$ .

Let us next discuss the choice of  $\underline{Z}_{\text{safe}}$ . Assume that the constraint (7b) is enforced over  $Z_{\text{safe}}$  instead of  $\underline{Z}_{\text{safe}}$ . In this case, the constraints (7b) and (7c), as well as the condition in (8) of Proposition 1, may be incompatible, leading to infeasibility of the optimization problem (7). This incompatibility arises from the fact that we are simultaneously asking for the value of  $h(x)$  to vary from  $\gamma_{\text{safe}}$  to  $\gamma_{\text{unsafe}}$  over a short distance of  $\epsilon + \epsilon_{\mathcal{N}}$  or even less while having a small Lipschitz constant. In particular, as posed, the constraints require that  $|h(x_s) - h(x_u)| \geq \gamma_{\text{safe}} + \gamma_{\text{unsafe}}$  for  $x_s \in Z_{\text{safe}}$  and  $x_u \in Z_{\mathcal{N}}$  safe and unsafe samples, respectively, but the sampling requirements ( $Z_{\text{safe}}$  and  $Z_{\mathcal{N}}$  being  $\epsilon$  and  $\epsilon_{\mathcal{N}}$ -nets of  $\mathcal{D}$  and  $\mathcal{N}$ , respectively) imply that  $\|x_s - x_u\| \leq \epsilon_{\mathcal{N}} + \epsilon$  for at least some pair  $(x_s, x_u)$ , which in turn implies that

$$\text{Lip}_h(\mathcal{B}_{\epsilon+\epsilon_{\mathcal{N}}}(x_u)) \geq \frac{|h(x_s) - h(x_u)|}{\|x_s - x_u\|} \geq \frac{\gamma_{\text{safe}} + \gamma_{\text{unsafe}}}{\epsilon_{\mathcal{N}} + \epsilon}.$$

Since  $\text{Lip}_h(\mathcal{B}_{\epsilon+\epsilon_{\mathcal{N}}}(x_u)) \approx \text{Lip}_h(\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_u))$ , the local Lipschitz constant  $\text{Lip}_h(\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_u))$  may get too large if  $\gamma_{\text{safe}}$  and  $\gamma_{\text{unsafe}}$  are chosen to be too large, and we may exceed the required upper bound of  $\gamma_{\text{unsafe}}/\epsilon_{\mathcal{N}}$  as per (8). We address this issue as follows: for fixed  $\gamma_{\text{safe}}$ ,  $\gamma_{\text{unsafe}}$ , and desired Lipschitz constant  $L_h < \gamma_{\text{unsafe}}/\epsilon_{\mathcal{N}}$ , we define

$$\underline{Z}_{\text{safe}} = \left\{ x_i \in Z_{\text{safe}} \mid \inf_{x \in Z_{\mathcal{N}}} \|x - x_i\| \geq \frac{\gamma_{\text{safe}} + \gamma_{\text{unsafe}}}{L_h} \right\} \quad (9)$$

which corresponds to a subset of admissible safe states, i.e.,  $\underline{Z}_{\text{safe}} \subset Z_{\text{safe}}$ . Intuitively, this introduces a buffer region across which  $h(x)$  can vary in value from  $\gamma_{\text{safe}}$  to  $-\gamma_{\text{unsafe}}$  for the desired Lipschitz constant  $L_h$ . Enforcing (7b) over  $\underline{Z}_{\text{safe}}$  allows for smoother functions  $h(x)$  to be learned at the expense of a smaller invariant safe set  $\mathcal{C}$ .

Note finally that, if  $h(x)$  is such that  $\mathcal{C} \subset \mathcal{D}$  (i.e., under the conditions in Proposition 1), then  $Y(\mathcal{C}) \subseteq Y(\mathcal{D}) = \mathcal{Y}$  by definition of  $\mathcal{Y}$  so that  $\mathcal{Y} \supseteq Y(\mathcal{C})$  as required in Theorem 1.

### 5.3.2 Increasing the volume of $\mathcal{C}$

We next explain how to avoid a disconnected learned safe set  $\mathcal{C}$ , which would degrade control performance, and show how to increase the volume of  $\mathcal{C}$ . Recall  $\underline{Z}_{\text{safe}}$  from the previous subsection and define

$$\underline{\mathcal{D}} := \bigcup_{x_i \in \underline{Z}_{\text{safe}}} \mathcal{B}_{\epsilon}(x_i).$$

Note that  $\underline{Z}_{\text{safe}}$  is an  $\epsilon$ -net of  $\underline{\mathcal{D}}$  by definition. Note also that  $\underline{\mathcal{D}}$  is a subset of  $\mathcal{D}$ , i.e.,  $\underline{\mathcal{D}} \subseteq \mathcal{D}$ . Following a similar argument as in Proposition 1, we next show conditions under which  $h(x) \geq 0$  for all  $x \in \underline{\mathcal{D}}$ .

**Proposition 2.** *Let  $h(x)$  be Lipschitz continuous with local Lipschitz constant  $\text{Lip}_h(\mathcal{B}_{\epsilon}(x_i))$  within the set  $\mathcal{B}_{\epsilon}(x_i)$  for  $x_i \in \underline{Z}_{\text{safe}}$ . Let  $\gamma_{\text{safe}} > 0$  and let*

$$\epsilon \leq \frac{\gamma_{\text{safe}}}{\text{Lip}_h(\mathcal{B}_{\epsilon}(x_i))} \quad (10)$$

for all  $x_i \in \underline{Z}_{\text{safe}}$ . Then, the constraint (7b) ensures that  $h(x) \geq 0$  for all  $x \in \underline{\mathcal{D}}$ .

The previous result can be used to guarantee that the set  $\mathcal{C}$  defined in equation (5) contains the set  $\underline{\mathcal{D}}$ , i.e.,  $\underline{\mathcal{D}} \subseteq \mathcal{C}$ . Hence, the set  $\underline{\mathcal{D}}$  can be seen as the minimum volume of the set  $\mathcal{C}$  that we can guarantee. Note in particular that, under the conditions in Propositions 1 and 2, it holds that  $\mathcal{C}$  is such that  $\underline{\mathcal{D}} \subseteq \mathcal{C} \subset \mathcal{D} \subseteq \mathcal{S}$ .

### 5.3.3 Guaranteeing that $h(x)$ is a ROCBF

The conditions in the previous subsection guarantee that the level-sets of the learned function  $h(x)$  satisfy the desired geometric safety properties. We now derive conditions that ensure that  $h(x)$  is a ROCBF, i.e., that the constraint (4) is also satisfied. Note that to verify (4), it suffices to show that there exists a control law such that  $U(y, t) \in K_{\text{ROCBF}}(y, t)$ . Our approach is to use the control inputs provided by the expert demonstrations.

We first show that the  $y$  components of  $Z_{\text{dyn}}$  cover the set of admissible output measurements  $\mathcal{Y}$  uniformly dense.

**Lemma 1.** *Let  $\bar{\epsilon} := \text{Lip}_Y(\overline{\mathcal{D}})(\epsilon + \bar{\Delta}_X)$  where  $\text{Lip}_Y(\overline{\mathcal{D}})$  is the local Lipschitz constant of the function  $Y$  within the set  $\overline{\mathcal{D}}$*

$$\overline{\mathcal{D}} := \mathcal{D} \oplus \mathcal{B}_{2\bar{\Delta}_X}(0)$$

where  $\bar{\Delta}_X := \sup_{y \in \mathcal{Y}} \Delta_X(y)$ .<sup>4</sup> Then the  $y$  components of  $Z_{\text{dyn}}$  form an  $\bar{\epsilon}$ -net of  $\mathcal{Y}$ .

For fixed  $u \in \mathcal{U}$  and  $t \in \mathbb{R}_{\geq 0}$ , the function  $q(u, y, t)$  from (6) is Lipschitz continuous<sup>5</sup> with local Lipschitz constant denoted by  $\text{Lip}_q(u, \mathcal{Y}, t)$  within the set  $\mathcal{Y} \subseteq \mathbb{R}^p$ . Let us also denote by  $\text{Bnd}_q(u, y)$  the bound in the difference of  $q$  for all times  $t$ . More formally, for each  $\bar{y} \in \mathcal{B}_{\bar{\epsilon}}(y)$ , it holds that

$$|q(u, \bar{y}, t') - q(u, \bar{y}, t'')| \leq \text{Bnd}_q(u, y), \forall t', t'' \geq 0. \quad (11)$$

The bound  $\text{Bnd}_q(u, y)$  exists and can be obtained as all components of  $q(u, y, t)$  are bounded in  $t$ . This is a natural assumption to obtain formal guarantees on our learned ROCBF  $h(x)$  from a finite data set  $Z_{\text{dyn}}$  since it is not possible to sample the time domain  $\mathbb{R}_{\geq 0}$  densely with a finite number of samples. These bounds can be neglected, i.e.,  $\text{Bnd}_q(u, y) = 0$ , when the system (1) is independent of  $t$ .

We next provide conditions under which: 1) for each  $(y, t) \in \mathcal{Y} \times \mathbb{R}_{\geq 0}$ , there exists a  $u \in \mathcal{U}$  such that  $q(u, y, t) \geq 0$ , and 2) the learned function  $h(x)$  is a ROCBF.

**Proposition 3.** *For  $(u_i, y_i, t_i) \in Z_{\text{dyn}}$ , let: 1)  $q(u_i, y, t_i)$  be Lipschitz continuous in  $y$  with local Lipschitz constant  $\text{Lip}_q(u_i, \mathcal{B}_{\bar{\epsilon}}(y_i), t_i)$  within the set  $\mathcal{B}_{\bar{\epsilon}}(y_i)$ , and 2)  $\text{Bnd}_q(u_i, y_i)$  be as in (11). Furthermore, let  $\gamma_{\text{dyn}} > 0$  and*

$$\bar{\epsilon} \leq \frac{\gamma_{\text{dyn}} - \text{Bnd}_q(u_i, y_i)}{\text{Lip}_q(u_i, \mathcal{B}_{\bar{\epsilon}}(y_i), t_i)} \quad (12)$$

for all  $(u_i, y_i, t_i) \in Z_{\text{dyn}}$ . Then, the constraint (7d) ensures that, for each  $(y, t) \in \mathcal{Y} \times \mathbb{R}_{\geq 0}$ , there exists a  $u \in \mathcal{U}$  such that  $q(z, u, t) \geq 0$ . If additionally  $\bar{L}_j \geq \text{Lip}_{B_j}(\overline{\mathcal{D}}, t)$  for each  $j \in \{1, 2, 3\}$ , then  $h(x)$  is a ROCBF.

---

<sup>4</sup>The set  $\overline{\mathcal{D}}$  is equivalent to the set of admissible safe states  $\mathcal{D}$  enlarged by a ball of size  $2\bar{\Delta}_X$  where  $\bar{\Delta}_X$  is the maximum attainable value of  $\Delta_X(y)$  on the set of admissible output measurements  $\mathcal{Y}$ . Using  $\text{Lip}_Y(\overline{\mathcal{D}})$ , it is guaranteed that the Lipschitz constant of  $Y$  on a sufficiently large set is considered.

<sup>5</sup>As the function  $h(x)$  is twice continuously differentiable, we immediately have that  $\nabla h(x)$  is locally Lipschitz continuous over the bounded domain  $\mathcal{D}$ . Also note that  $\hat{F}$ ,  $\hat{G}$ ,  $\Delta_F$ ,  $\Delta_G$ ,  $\alpha$ ,  $h(x)$ ,  $\hat{X}$ , and  $\Delta_X$  are Lipschitz continuous.

In summary, Proposition 3 says that a larger Lipschitz constant of the function  $q$  requires a larger margin  $\gamma_{\text{dyn}}$  and/or a smaller  $\bar{\epsilon}$ , i.e., a finer net of safe datapoints as indicated by  $\epsilon$  and/or a reduction in the measurement map error  $\Delta_X$ .

The next theorem, which follows immediately from the previous results and which is provided without proof, summarizes the set of sufficient conditions guaranteeing that a function  $h(x)$  obtained from (7) is a ROCBF on  $\mathcal{Y}$ .

**Theorem 2.** *Let  $h(x)$  be a twice continuously differentiable function. Let the sets  $\mathcal{S}$ ,  $\mathcal{C}$ ,  $\mathcal{Y}$ ,  $\underline{\mathcal{D}}$ , and  $\mathcal{N}$  as well as the data-sets  $Z_{\text{safe}}$ ,  $Z_{\text{dyn}}$ , and  $Z_{\mathcal{N}}$  be defined as above. Suppose that  $Z_{\mathcal{N}}$  forms an  $\epsilon$ -net of  $\mathcal{N}$  and that the conditions (8), (10), and (12) are satisfied. Assume also that  $\bar{L}_j \geq \text{Lip}_{B_j}(\bar{\mathcal{D}}, t)$  for each  $j \in \{1, 2, 3\}$ . If  $h(x)$  satisfies the constraints (7b), (7c), and (7d), then  $h(x)$  is a ROCBF on  $\mathcal{Y}$  and it holds that the set  $\mathcal{C}$  is non-empty and such that  $\underline{\mathcal{D}} \subseteq \mathcal{C} \subseteq \mathcal{S}$ .*

## 6 Algorithmic Implementation

In this section, we present the algorithmic implementation of the previously presented results. We also discuss several aspects related to solving the unconstrained optimization problem (7), the construction of the involved datasets, and estimating Lipschitz constants of the functions  $h$  and  $q$ .

### 6.1 The Algorithm

We present upfront Algorithm 1 that summarizes the steps to learn safe ROCBFs  $h(x)$  from expert demonstrations and explain the details in the remainder.

---

#### Algorithm 1 Learning ROCBF from Expert Demonstrations

---

**Input:** Set of expert demonstrations  $Z_{\text{dyn}}$ ,  
 1: system model  $(\hat{F}, \hat{G}, \hat{X}, \Delta_F, \Delta_G, \Delta_X)$ ,  
 2: hyperparameters  $(\alpha, \gamma_{\text{safe}}, \gamma_{\text{unsafe}}, \gamma_{\text{dyn}}, L_h, \bar{L}_1, \bar{L}_2, \bar{L}_3, k, \eta)$   
**Output:** Safe ROCBF  $h(x)$   
 3:  $Z_{\text{safe}} \leftarrow \cup_{(u_i, y_i, t_i) \in Z_{\text{dyn}}} \hat{X}(y_i)$  # Safe datapoints  
 4:  $Z_{\mathcal{N}} \leftarrow \text{BPD}(Z_{\text{safe}}, k, \eta)$  # Unsafe datapoints obtained by boundary point detection (BPD) in Algorithm 2.  
 5:  $Z_{\text{safe}} \leftarrow Z_{\text{safe}} \setminus Z_{\mathcal{N}}$   
 6:  $\underline{Z}_{\text{safe}} \leftarrow$  according to (9)  
 7:  $h(x) \leftarrow$  solution of (13) # relaxation of the constrained optimization problem in (7)  
 8: **if** constraints (7b)-(7d) or constraints (8), (10), (12) are not satisfactorily satisfied **then**  
 9:   Modify hyperparameters and start from line 3  
 10: **end if**

---

In summary, we first construct the set of safe datapoints  $Z_{\text{safe}}$  in line 3 based on the expert demonstrations  $Z_{\text{dyn}}$ . In line 4, we construct the set of as unsafe labeled datapoints  $Z_{\mathcal{N}}$  from  $Z_{\text{safe}}$ , i.e.,  $Z_{\mathcal{N}} \subseteq Z_{\text{safe}}$ , by identifying boundary points in  $Z_{\text{safe}}$  and labeling them as unsafe datapoints (details can be found in Section 6.2). We then re-define  $Z_{\text{safe}}$  by removing the as unsafe labeled datapoints  $Z_{\mathcal{N}}$  from  $Z_{\text{safe}}$  in line 5. Following our discussion in Section 5.3.1, we obtain  $\underline{Z}_{\text{safe}}$  as in (9) in line 6. In line 7, we attempt to solve the unconstrained optimization problem defined in (7) by an unconstrained relaxation thereof as defined in (13) and further discussed in Section

6.3. We then check whether or not the constraints (7b)-(7d) and the constraints (8), (10), (12) are satisfactorily satisfied by the learned function  $h(x)$ . To check these constraints, we need to be able to estimate Lipschitz constants as detailed in Section 6.4. Note also that we loosely use the term ‘‘satisfactorily’’ here as this may depend on the actual application and satisfaction of all constraints may in practice not be possible. If the constraints are not satisfactorily satisfied, the set of hyperparameters is tweaked based on the observed violation and the process is repeated (line 9).

## 6.2 Construction of the Datasets

Due to the conditions in (8), (10), and (12), the first requirement on the data is that the datasets  $Z_{\text{safe}}$  and  $Z_{\mathcal{N}}$  are dense in a geometric sense requiring that  $\epsilon$  and  $\epsilon_{\mathcal{N}}$  are small. It is also required that  $Z_{\mathcal{N}}$  is an  $\epsilon_{\mathcal{N}}$ -net of  $\mathcal{N}$ . In order to construct the  $\epsilon_{\mathcal{N}}$ -net  $Z_{\mathcal{N}}$  of  $\mathcal{N}$ , a simple randomized algorithm which repeatedly uniformly samples from  $\mathcal{N}$  works with high probability (see, for example, [53]). Hence, as long as we can efficiently sample from  $\mathcal{N}$ , e.g., when  $\mathcal{N}$  is a basic primitive set or has an efficient set-membership oracle, uniform sampling or gridding methods are viable strategies. However, as this is in general not possible, we instead use a boundary point detection algorithm in Algorithm 1 as explained next.

As opposed to above where  $\mathcal{N}$  is given and we obtain samples defining  $Z_{\mathcal{N}}$ , the idea is to obtain  $Z_{\mathcal{N}}$  from  $Z_{\text{safe}}$  and then obtain  $\mathcal{N}$  given  $\epsilon_{\mathcal{N}}$ . To perform this step efficiently, our approach is to detect geometric boundary points of the set  $Z_{\text{safe}}$ . This subset of boundary points is labeled as  $Z_{\mathcal{N}}$ , while we re-define  $Z_{\text{safe}}$  to exclude the boundary points  $Z_{\mathcal{N}}$ . Crucially, we need to obtain the set  $Z_{\mathcal{N}}$  with a corresponding  $\epsilon_{\mathcal{N}}$  such that the set  $\mathcal{N}$  surrounds the set  $\mathcal{D}$ . Therefore, one can artificially sample additional datapoints in proximity of  $Z_{\mathcal{N}}$  and add these to the set  $Z_{\mathcal{N}}$ .

We detect boundary points in  $Z_{\text{safe}}$  based on the concept of reverse  $k$ -nearest neighbors, see e.g., [54]. The main idea is that boundary points typically have fewer reverse  $k$ -nearest neighbors than interior points. In summary, for  $k > 0$  we find the  $k$ -nearest neighbors of each datapoint  $x_i \in Z_{\text{safe}}$ . Then, we find the reverse  $k$ -nearest neighbors of each datapoint  $x_i \in Z_{\text{safe}}$ , that is, we find the datapoints  $x'_i \in Z_{\text{safe}}$  that have  $x_i$  as their  $k$ -nearest neighbor. Finally, we choose a threshold  $\eta > 0$  and label all datapoints  $x_i \in Z_{\text{safe}}$  as a boundary point whose cardinality of reverse  $k$ -nearest neighbors is below  $\eta$ .

---

**Algorithm 2** Boundary Point Detection - BPD( $Z_{\text{safe}}, k, \eta$ )

---

**Input:** Set of safe states  $Z_{\text{safe}}$ , number of nearest neighbors  $k$ , neighbor threshold  $\eta > 0$

**Output:** Set of boundary points, i.e., the set of unsafe labeled states  $Z_{\mathcal{N}}$

- 1:  $M \leftarrow \text{compute\_pairwise\_dists}(Z_{\text{safe}})$  # compute pairwise distances between elements in  $Z_{\text{safe}}$
  - 2:  $kNN_i \leftarrow \text{compute\_k\_nearest\_neighbors}(M)$  # compute  $k$ -nearest neighbors of  $x_i \in Z_{\text{safe}}$
  - 3:  $RkNN_i \leftarrow \text{compute\_reverse\_k\_nearest\_neighbors}(kNN)$  # compute reverse  $k$ -nearest neighbors of  $x_i \in Z_{\text{safe}}$
  - 4:  $z_i \leftarrow \mathbb{1}(RkNN_i \leq \eta)$  # threshold  $RkNN_i$  by  $\eta$
- 

The detailed boundary point detection algorithm is summarized in Algorithm 2. In line 1, we first compute the pairwise distances between each of the  $N_1$  safe datapoints in  $Z_{\text{safe}}$ ; the result is a symmetric  $N_1 \times N_1$  matrix  $M$ , where the element at position  $(i, j)$  in  $M$  represents the pairwise distance between the states  $x_i$  and  $x_j$ , i.e.,  $M_{ij} := \|x_i - x_j\|$ . Next, we calculate the  $k$ -nearest neighbors of each  $x_i$  in line 2, denoted by  $kNN_i$ , as the set of indices corresponding to the  $k$

smallest column entries in the  $i$ th row of  $M$ . In line 3, we calculate the reverse  $k$ -nearest neighbors of each  $x_i$  as  $RkNN_i := |\{x_j \in Z_{\text{safe}} | x_i \in kNN_j\}|$ . Finally in line 4, we threshold each  $RkNN_i$  by  $\eta$ , i.e.,  $z_i := \mathbb{1}(RkNN_i \leq \eta)$  where  $\mathbb{1}$  is the indicator function. We obtain a tuple  $(z_1, \dots, z_{N_1}) \in \mathbb{R}^{N_1}$ , where the indices  $i$  that are set to one correspond to states  $x_i$  that the algorithm identifies as boundary points.

As of yet, we have not specified  $\epsilon$  and  $\epsilon_{\mathcal{N}}$  that are needed to check the constraints (8), (10), and (12). While the value of  $\epsilon$  merely defines  $\mathcal{D}$  and determines the size of the set  $\mathcal{C}$ , the value of  $\epsilon_{\mathcal{N}}$  is important as  $\mathcal{N}$  should fully enclose  $\mathcal{D}$ . This imposes an implicit lower bound on  $\epsilon_{\mathcal{N}}$  to guarantee safety. One possible way to get an estimate of  $\epsilon_{\mathcal{N}}$  is calculate the distance of each datapoint to the closest datapoint in  $Z_{\mathcal{N}}$ , respectively. Then, taking the maximum or an average over these values gives an estimate of  $\epsilon_{\mathcal{N}}$ .

Finally, let us discuss what behavior the expert demonstrations in  $Z_{\text{dyn}}$  should exhibit. At a high level, our results state that if a smooth function  $h(x)$  can be found that satisfies the constraints (7b)-(7d) over a sufficiently fine sampling of the state-space (as specified in (8), (10), and (12)), then the resulting function is a safe RCBF. We focus here on the RCBF constraint (4), which must be verified to hold for some  $u \in \mathcal{U}$ , by using the expert demonstration  $(u_i, y_i, t_i)$  in the constraint (7d). In particular, the more transverse the vector field of the input dynamics  $\langle \hat{G}(\hat{X}(y_i), t_i), u_i \rangle$  is to the level sets of the function  $h(\hat{X}(y_i))$  (i.e., the more parallel it is to the inward pointing normal  $\nabla h(\hat{X}(y_i))$ ), the larger the inner-product term in constraint (7d) will be without increasing the Lipschitz constant of  $h(x)$ . In words, this says that the expert demonstrations should demonstrate how to move away from the unsafe set.

### 6.3 Solving the Constrained Optimization Problem

Some remarks are in place with respect to solving the optimization problem (7). If the extended class  $\mathcal{K}$  function  $\alpha$  is linear and the set  $\mathcal{H}$  is parameterized as  $\mathcal{H} := \{\langle \phi(x), \theta \rangle | \theta \in \Theta\}$  where  $\Theta \subseteq \mathbb{R}^l$  is a convex set and  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^l$  is a known but nonlinear function, then the optimization problem (7) is convex and can be solved efficiently. Note here in particular that  $\|\nabla h(x)\|_{\star}$  is convex in  $\theta$  since 1)  $\nabla h(x)$  is linear in  $\theta$ , 2) norms are convex functions, and 3) composition of a convex with a linear function preserves convexity. Note that very rich function classes such as infinite dimensional reproducing kernel Hilbert spaces from statistical learning theory can be approximated to arbitrary accuracy as such a  $\mathcal{H}$  [55].

In the more general case when  $\mathcal{H} := \{h(x, \theta) | \theta \in \Theta\}$ , such as when  $h(x, \theta)$  is a deep neural network or when  $\alpha$  is a general nonlinear function, the optimization problem (7) is nonconvex. Due to the computational complexity of general nonlinear constrained programming, we propose an unconstrained relaxation of the optimization problem (7) that can be solved efficiently in practice by first order gradient based methods. Let  $[r]_+ := \max\{r, 0\}$  for  $r \in \mathbb{R}$ . Our unconstrained relaxation results in the following optimization problem:

$$\underset{\theta \in \Theta}{\operatorname{argmin}} \|\theta\|^2 + \lambda_s \sum_{x_i \in Z_{\text{safe}}} [\gamma_{\text{safe}} - h(x_i, \theta)]_+ + \lambda_u \sum_{x_i \in Z_{\mathcal{N}}} [h(x_i, \theta) + \gamma_{\text{unsafe}}]_+, \quad (13a)$$

$$+ \lambda_d \sum_{(u_i, y_i, t_i) \in Z_{\text{dyn}}} [\gamma_{\text{dyn}} - q(u_i, y_i, t_i, \theta)]_+ \quad (13b)$$

where the function  $q(u, y, t, \theta)$  is the same as  $q(u, y, t)$  in (6), but now defined for the parametrized function  $h(x, \theta)$  instead of  $h(x)$ . The positive parameters  $\lambda_s$ ,  $\lambda_u$ , and  $\lambda_d$  allow us to trade off the

relative importance of each of the terms in the optimization. While the unconstrained optimization problem (13) of the original optimization problem (7) is in general a nonconvex optimization problem, it can be solved efficiently in practice with stochastic first-order gradient methods such as Adam or stochastic gradient descent.

On another note, the optimization problem (7) approximates the supremum over control inputs  $u \in \mathcal{U}$  in the constraint (4) with the expert actions  $u_i$  in (7d). While this brings computational benefits, this may prove conservative. For the special case when  $\Delta_X(y) = \Delta_F(x, t) = \Delta_G(x, t) = 0$ , i.e., the system dynamics as well as the output measurement map are perfectly known, the supremum over the control input  $u \in \mathcal{U}$  in constraint (4) may admit a closed-form expression. For example, when  $\mathcal{U}$  is a unit  $\|\cdot\|$ -norm ball, the function  $B(x, t, u)$  in the left-hand side of (7d) reduces to

$$\langle \nabla h(x), \hat{F}(x, t) \rangle + \|\langle \nabla h(x), \hat{G}(x, t) \rangle\|_* + \alpha(h(x)).$$

This simplifies the constraint (7d) in the optimization problem (7), and in particular eliminates the dependency on  $u_i$ . Nevertheless, the availability of expert demonstrations is valuable as they indicate that a safe action exists, and we therefore expect a feasible ROCBF  $h$  and control action  $u$  to exist.

## 6.4 Lipschitz Constants

Let us again focus on the conditions (8), (10), and (12) that contain Lipschitz constants of the functions  $h$  and  $q$ . As described earlier, because we assume that functions in  $\mathcal{H}$  are twice continuously differentiable and we restrict ourselves to a compact domain  $\mathcal{N} \cup \mathcal{D}$ , we immediately have that  $h$  and  $\nabla h$  are both uniformly Lipschitz over  $\mathcal{N} \cup \mathcal{D}$ . We show here two examples of  $\mathcal{H}$  where it is computationally efficient to estimate an upper bound on the Lipschitz constants of functions  $h \in \mathcal{H}$ .

In the case of random Fourier features with  $\ell$  random features, where  $h(x) := \langle \phi(x), \theta \rangle$  and  $\phi(x) \in \mathbb{R}^\ell$  is

$$\phi(x) := \sqrt{2/\ell}(\cos(\langle x, w_1 \rangle + b_1), \dots, \cos(\langle x, w_\ell \rangle + b_\ell)),$$

then we can analytically compute upper bounds as follows. First, we have by the Cauchy-Schwarz inequality  $|h(x_1) - h(x_2)| \leq \|\phi(x_1) - \phi(x_2)\|_2 \|\theta\|_2$ . To bound  $\|\phi(x_1) - \phi(x_2)\|_2$ , we bound the spectral norm of the Jacobian  $D\phi(x)$ , which is a matrix where the  $i$ -th row is  $-\sqrt{2/\ell} \sin(\langle x, w_i \rangle + b_i) w_i^\top$ . Let  $s_i := \sin(\langle x, w_i \rangle + b_i)$  and observe that  $\|D\phi(x)\| = \sqrt{2/\ell} \sup_{\|v\|_2=1} (\sum_{i=1}^{\ell} s_i^2 \langle w_i, v \rangle^2)^{1/2} \leq \sqrt{2/\ell} \sup_{\|v\|_2=1} (\sum_{i=1}^{\ell} \langle w_i, v \rangle^2)^{1/2} = \sqrt{2/\ell} \|\mathcal{W}\|$ , where  $\mathcal{W}$  is a matrix with the  $i$ -th row equal to  $w_i$ . While the bound  $\sqrt{2/\ell} \|\mathcal{W}\|$  can be used in computations, we can further understand order-wise scaling of the bound as follows. For random Fourier features corresponding to the popular Gaussian radial basis function kernel,  $w_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2 I)$  where  $\sigma^2$  is the (inverse) bandwidth of the Gaussian kernel. Therefore, by standard results in non-asymptotic random matrix theory [53], we have that  $\|\mathcal{W}\| \leq \sigma(\sqrt{\ell} + \sqrt{n} + \sqrt{2 \log(1/\delta)})$  with a probability of at least  $1 - \delta$ . Combining these calculations, we have that the Lipschitz constant of  $h$  can be bounded by  $\sqrt{2\sigma^2(1 + \sqrt{n/\ell} + \sqrt{(2/\ell) \log(1/\delta)})} \|\theta\|_2$  with a probability of at least  $1 - \delta$ .

We now bound the Lipschitz constant of the gradient  $\nabla h(x) = D\phi(x)^\top \theta$ . We do this by bounding the spectral norm of the Hessian  $\nabla^2 h(x) = -\sqrt{2/\ell} \sum_{i=1}^{\ell} c_i \theta_i w_i w_i^\top$ , with  $c_i = \cos(\langle x, w_i \rangle + b_i)$ .



(a) Training course. (b) Test course. (c) Downsampled RGB camera image.

Figure 4: Simulation environment in CARLA. The cars are tracking desired reference paths on different courses. Left: The **training course** from which training data, during a left turn, was generated to train and test the ROCBF. Middle: An unknown **test course** on which the learned ROCBF is tested. Right: Downsampled RGB dashboard camera image during the left turn on the training course.

$b_i$ ). A simple bound is  $\|\nabla^2 h(x)\| \leq \sqrt{2/\ell} \|\theta\|_\infty \|\mathcal{W}\|^2 \leq 3\sqrt{2} \|\theta\|_\infty \sigma^2 (\ell + n + 2 \log(1/\delta)) / \sqrt{\ell}$ , where the last inequality holds with a probability of at least  $1 - \delta$ . Note that, based on the calculated Lipschitz constants of  $h$  and  $\nabla h$ , a Lipschitz constant of  $q$  can be easily obtained.

When  $h(x)$  is a DNN, accurately estimating the Lipschitz constant is more involved. In general, the problem of exactly computing the Lipschitz constant of  $h$  is known to be NP-hard [56]. Notably, because most commonly-used activation functions  $\phi$  are known to be 1-Lipschitz (e.g. ReLU, tanh, sigmoid), a naive upper bound on the Lipschitz constant of  $h$  is given by the product of the norms of the weight matrices; that is,  $\text{Lip}_h \leq \prod_k \|\mathcal{W}^k\|$ . However, this bound is known to be quite loose [57]. Recently, the authors of [57] proposed a semidefinite-programming based approach to efficiently compute an accurate upper bound on  $\text{Lip}_h$ . In particular, this approach relies on incremental quadratic constraints to represent the couplings between pairs of neurons in the neural network  $h$ . On the other hand, there are relatively few results that provide accurate upper bounds for the Lipschitz constant of the gradient of  $h$  when  $h$  is a neural network. While ongoing work looks to extend the results from [57] to compute upper bounds on  $\text{Lip}_{\nabla h}$ , to the best of our knowledge, the only general method for computing an upper bound on  $\text{Lip}_{\nabla h}$  is through post-hoc sampling [58].

## 7 Simulations

We construct a safe ROCBF for a car driving on a highway within the autonomous driving simulator CARLA [3], see Fig. 4. In particular, our goal is to learn a ROCBF for the lateral control of the car, i.e., a lane keeping controller, while we use a built-in controller for longitudinal control. Lane keeping in CARLA is achieved by tracking a set of predefined waypoints. The code for our simulations and videos of the car under the learned safe ROCBFs is available at

<https://github.com/unstable-zeros/learning-rocbfs>.

As we have no direct access to the system dynamics of the car, we identify a system model. The model for longitudinal control is estimated from data and consists of the velocity  $v$  of the car and the integrator state  $d$  of the PID. The identified longitudinal model of the car is

$$\dot{v} = -1.0954v - 0.007v^2 - 0.1521d + 3.7387$$

$$\dot{d} = 3.6v - 20.$$

For the lateral control of the car, we consider a bicycle model

$$\begin{aligned}\dot{p}_x &= v \cos(\theta), \\ \dot{p}_y &= v \sin(\theta), \\ \dot{\theta} &= v/L \tan(\delta),\end{aligned}$$

where  $p_x$  and  $p_y$  denote the position of the car in a global coordinate frame,  $\theta$  is the heading angle of the car with respect to a global coordinate frame, and  $L := 2.51$  is the distance between the front and the rear axles of the car. The control input here is the steering angle  $\delta$  that we would like to design such that the car is tracking predefined waypoints that are provided by CARLA. Note that treating  $u := \tan(\delta)$  as the control input gives us a control affine system as in (1).

To be able to learn a ROCBF in a data efficient manner, we convert the above lateral model (defined in a global coordinate frame) into a local coordinate frame. We do so relatively to the waypoints that the car has to follow to stay within the lane. We consider the cross-track error  $c_e$  of the car. In particular, let  $wp_1$  be the waypoint that is closest to the car and let  $wp_2$  be the waypoint proceeding  $wp_1$ . Then the cross-track error is defined as  $c_e := \|w\| \sin(\theta_w)$  where  $w \in \mathbb{R}^2$  is the vector pointing from  $wp_1$  to the car and  $\theta_w$  is the angle between  $w$  and the vector pointing from  $wp_1$  to  $wp_2$ . We further consider the error angle  $\theta_e := \theta - \theta_t$  where  $\theta_t$  is the angle between the vector pointing from  $wp_1$  to  $wp_2$  and the global coordinate frame. The simplified local model is then defined as

$$\begin{aligned}\dot{c}_e &= v \sin(\theta_e), \\ \dot{\theta}_e &= v/2.51u - \dot{\theta}_t.\end{aligned}$$

In summary, we have the state  $x := [v \ d \ c_e \ \theta_e]^T$  and the control input  $u := \tan(\delta)$  as well as the external input, given during runtime, of  $\dot{\theta}_t$ . Consequently, let

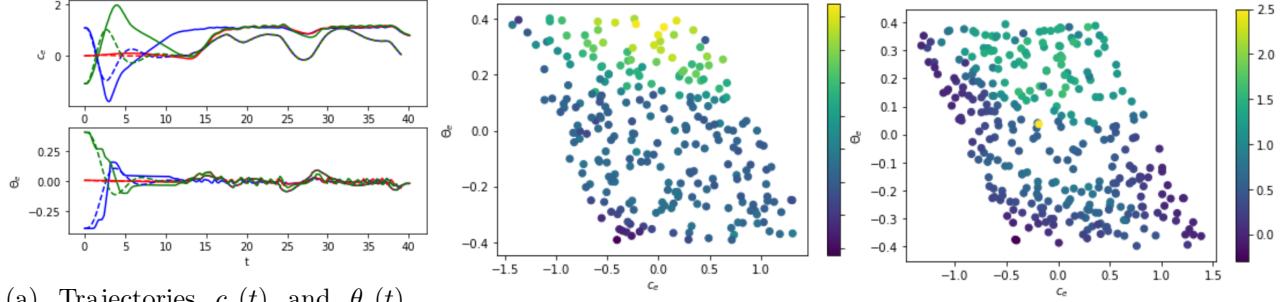
$$\hat{F} := \begin{bmatrix} -1.095v - 0.007v^2 - 0.152d + 3.74 \\ 3.6v - 20 \\ v \sin(\theta_e) \\ -\dot{\theta}_t \end{bmatrix} \quad \hat{G} := \begin{bmatrix} 0 \\ 0 \\ 0 \\ v/2.51 \end{bmatrix}$$

along with estimated error bounds  $\Delta_F(x, t) := 0.1$  and  $\Delta_G(x, t) = 0.1$ .

For collecting safe expert demonstrations  $Z_{\text{dyn}}$ , we use an “expert” PID controller  $u(x)$  that uses full state knowledge of  $x$ . Throughout this section, we use the parameters  $\alpha(r) := r$ ,  $\gamma_{\text{safe}} := \gamma_{\text{unsafe}} := 0.05$ , and  $\gamma_{\text{dyn}} := 0.01$  to train safe ROCBFs  $h(x)$ . For the boundary point detection algorithm in Algorithm 2, we select  $k := 200$  and  $\eta$  such that 40 percent of the points in  $Z_{\text{safe}}$  are labeled as boundary points.

## 7.1 State-based ROCBF

We first learn a ROCBF controller in the case that the state  $x$  is perfectly known, i.e., the model of the output measurement map is such that  $\hat{X}(y) = X(y) = x$  and the error is  $\Delta_X(y) := 0$ . The trained ROCBF  $h(x)$  is a two-layer DNN with 32 and 16 neurons per layer.



(a) Trajectories  $c_e(t)$  and  $\theta_e(t)$  for different initial conditions indicated by color. Solid and dashed lines correspond to the learned ROCBF and expert, respectively.

(b) Shown are 250 different initial conditions on the **training** course. The color legend encodes  $\max_t |c_e^{ROCBF}(t)| - \max_t |c_e^{Exp}(t)|$ .

(c) Shown are 300 different initial conditions on the **test course**. The color legend encodes  $\max_t |c_e^{ROCBF}(t)| - \max_t |c_e^{Exp}(t)|$ .

Figure 5: State-based ROCBF controller.

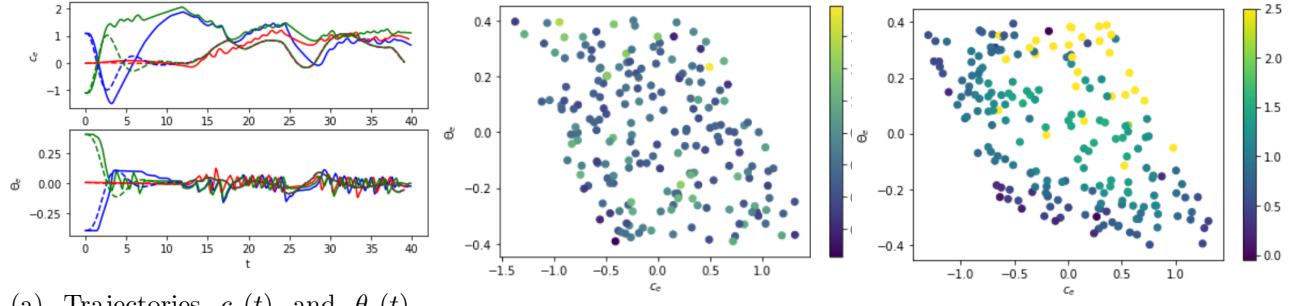
The safety controller applied to the car is then obtained as the solution of the convex optimization problem  $\min_{u \in \mathcal{U}} \|u\|$  subject to the constraint  $q(u, y, t) \geq 0$ . In Fig. 5a, example trajectories of  $c_e(t)$  and  $\theta_e(t)$  under this controller are shown. Solid lines indicate the learned ROCBF controller, while dashed lines indicate the expert PID controller for comparison. Colors in both subplots match the corresponding trajectories. The initial conditions of  $d(0)$  and  $v(0)$  are set to zero in all cases here, similar to all other plots in the remainder. Fig. 5b shows different initial conditions  $c_e(0)$  and  $\theta_e(0)$  and how the ROCBF controller performs relatively to the expert PID controller on the training course. In particular, each point in the plot indicates an initial condition from which system trajectories under both the ROCBF and expert PID controller are collected. The color map then shows

$$\max_t |c_e^{ROCBF}(t)| - \max_t |c_e^{Exp}(t)|$$

where  $c_e^{ROCBF}(t)$  and  $c_e^{Exp}(t)$  denote the cross-track errors under the ROCBF and expert PID controllers, respectively. Fig. 5c shows the same plot, but for the test course from which no data has been collected to train the ROCBF. In this plot, one ROCBF trajectory resulted in a collision as detected by CARLA. We assign by default a value of 2.5 in case of a collision (see the yellow point in Fig. 5c).

## 7.2 Perception-based ROCBF

We next learn a ROCBF in the case that  $y$  corresponds to images taken from an RGB camera mounted to the dashboard of the car. To train a perception map  $\hat{X}$ , we have resized the images as shown in Fig. 4c. We assume knowledge of  $\theta_e$ ,  $v$ , and  $d$ , while we estimate  $c_e$  from  $y$ , i.e.,  $x := [v \ d \ \hat{X}(y) \ \theta_e]^T$ . The architecture of  $\hat{X}$  is Resnet18, i.e., a convolutional neural network with 18 layers. The performance on the training data is shown in Fig. 7. Additionally, we set  $\Delta_X(y) := 0.1$  to account for potential estimation errors and we pick  $\bar{L}_1 := 1$ ,  $\bar{L}_2 + \bar{L}_3 := 1$ . The trained ROCBF  $h(x)$  is again a two-layer DNN with 32 and 16 neurons per layer. Figs. 6a-6c show the same plots as in the previous section and evaluate the ROCBF relatively to the expert PID controller. Importantly, note here that the expert PID controller uses state knowledge, while the



(a) Trajectories  $c_e(t)$  and  $\theta_e(t)$  for different initial conditions indicated by color. Solid and dashed lines correspond to the learned ROCBF and expert, respectively.

(b) Shown are 200 different initial conditions on the **training** course. The color legend encodes **course**. The color legend encodes  $\max_t |c_e^{ROCBF}(t)| - \max_t |c_e^{Exp}(t)|$ .

(c) Shown are 200 different initial conditions on the **test course**. The color legend encodes **course**. The color legend encodes  $\max_t |c_e^{ROCBF}(t)| - \max_t |c_e^{Exp}(t)|$ .

Figure 6: Perception-based ROCBF controller.

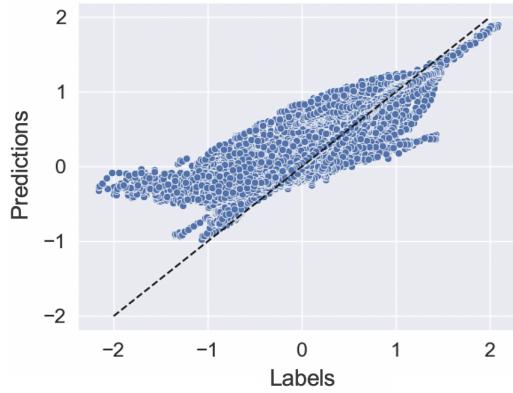


Figure 7: Performance of the perception map  $\hat{X}$  on training data.

ROCBF uses RGB images from the dashboard camera as inputs so that the relative gap between these two, as shown in Figs. 6b and 6c, becomes larger.

In Fig. 8, we further illustrate the level sets of the learned function  $h(x)$  within the  $c_e$ - $\theta_e$  plane for fixed values of  $v$  and  $d$ . Note that a visualization of the level sets in four dimensions is in general difficult.

## 8 Conclusion and Summary

In this paper, we have shown how safe control laws can be learned from expert demonstrations under system model and measurement map uncertainties. We first presented robust output control barrier functions (ROCBFs) as a means to enforce safety, which is here defined as the ability of a system to remain within a safe set using the notion of forward invariance. We then proposed an optimization problem to learn such ROCBFs from safe expert demonstrations, and presented verifiable conditions for the validity of the ROCBF. These conditions are stated in terms of the density of the data and on Lipschitz and boundedness constants of the learned function as well as

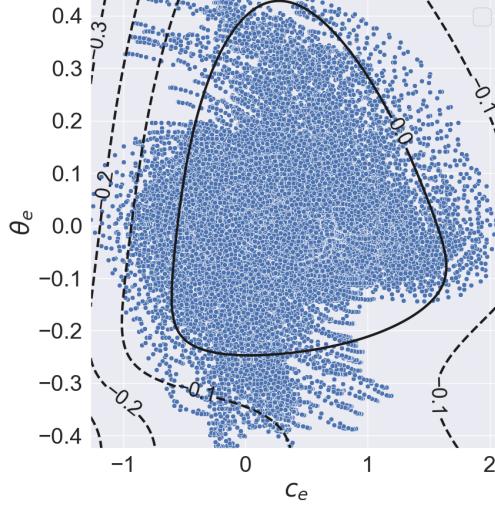


Figure 8: Illustration of the level sets of the learned function  $h(x)$  in the  $c_e$ - $\theta_e$  plane for fixed values of  $v$  and  $d$  for the perception-based ROCBF. The plot also shows all training datapoints projected onto the  $c_e$ - $\theta_e$  plane.

the models of the system dynamics and the measurement map. Finally, our simulation studies show how to learn safe control laws from RGB camera images within the autonomous driving simulator CARLA.

## A Proof of Theorem 1

Recall that  $f(t) := F(x(t), t)$ ,  $g(t) := G(x(t), t)$ ,  $y(t) := Y(x(t))$  and  $u(t) := U(y(t), t)$  according to (1c)-(1f) and let us define for convenience

$$\begin{aligned}\hat{f}(t) &:= \hat{F}(x(t), t) \\ \hat{g}(t) &:= \hat{G}(x(t), t) \\ \delta_F(t) &:= \Delta_F(x(t), t) \\ \delta_G(t) &:= \Delta_G(x(t), t) \\ \hat{x}(t) &:= \hat{X}(y(t)).\end{aligned}$$

Due to the chain rule and since  $U(y, t) \in K_{\text{ROCBF}}(y, t)$ , note that each solution  $x : \mathcal{I} \rightarrow \mathbb{R}^n$  to (1) under  $U(y, t)$  satisfies

$$B(\hat{x}(t), t, u(t)) \geq C(y(t), t, u(t))\Delta_X(y(t)). \quad (14)$$

Note now that the term  $\text{Lip}_{B_1}(\mathcal{X}(y(t)), t)\Delta_X(y(t))$  in the right-hand side of (14) can be written as

$$\begin{aligned}&\text{Lip}_{B_1}(\mathcal{X}(y(t)), t)\Delta_X(y(t)) \\ &\stackrel{(a)}{\geq} \text{Lip}_{B_1}(\mathcal{X}(y(t)), t)\|\hat{x}(t) - x(t)\|\end{aligned}$$

$$\stackrel{(b)}{\geq} |B_1(\hat{x}(t), t) - B_1(x(t), t)| \geq \\ B_1(\hat{x}(t), t) - B_1(x(t), t)$$

where (a) follows since  $x(t) \in \mathcal{X}(y(t))$  due to Assumption 2 and where (b) simply follows since  $\text{Lip}_{B_1}(\mathcal{X}(y(t)), t)$  is the local Lipschitz constant of the function  $B_1(x, t)$  within the set  $\mathcal{X}(y(t))$ . Next, the term  $\text{Lip}_{B_2}(\mathcal{X}(y(t)), t)\|u(t)\|_\star\Delta_X(y(t))$  in the right-hand side of (14) can similarly be written as

$$\begin{aligned} & \text{Lip}_{B_2}(\mathcal{X}(y(t)), t)\|u(t)\|_\star\Delta_X(y(t)) \\ & \geq \text{Lip}_{B_2}(\mathcal{X}(y(t)), t)\|\hat{x}(t) - x(t)\|\|u(t)\|_\star \\ & \geq \|B_2(\hat{x}(t), t) - B_2(x(t), t)\|\|u(t)\|_\star \\ & \stackrel{(c)}{\geq} \langle B_2(\hat{x}(t), t) - B_2(x(t), t), u(t) \rangle \end{aligned}$$

where (c) follows due to Hölder's inequality. For the last term  $\text{Lip}_{B_3}(\mathcal{X}(y(t)), t)\|u(t)\|\Delta_X(y(t))$  in the right-hand side of (14) it follows similarly that

$$\begin{aligned} & \text{Lip}_{B_3}(\mathcal{X}(y(t)), t)\|u(t)\|\Delta_X(y(t)) \\ & \geq \text{Lip}_{B_3}(\mathcal{X}(y(t)), t)\|\hat{x}(t) - x(t)\|\|u(t)\| \\ & \geq |B_3(\hat{x}(t), t) - B_3(x(t), t)|\|u(t)\| \\ & \geq (B_3(\hat{x}(t), t) - B_3(x(t), t))\|u(t)\|. \end{aligned}$$

From (14) and the definitions of the functions  $B(x, t, u)$  and  $C(y, t, u)$ , it hence follows that  $B(x(t), t, u(t)) \geq 0$ . Note next that the following holds

$$\begin{aligned} & B(x(t), t, u(t)) \geq 0 \\ & \Leftrightarrow \langle \nabla h(x(t)), \hat{f}(t) + \hat{g}(t)u(t) \rangle + \alpha(h(x(t))) \geq \|\nabla h(x(t))\|_\star(\delta_F(t) + \delta_G(t)\|u(t)\|) \\ & \stackrel{(d)}{\Rightarrow} \langle \nabla h(x(t)), f(t) + g(t)u(t) \rangle + \alpha(h(x(t))) \geq 0 \\ & \Leftrightarrow \dot{h}(x(t)) \geq -\alpha(h(x(t))) \end{aligned} \tag{15}$$

where the implication in (d) follows since

$$\begin{aligned} & \|\nabla h(x(t))\|_\star(\delta_F(t) + \delta_G(t)\|u(t)\|) \\ & \stackrel{(e)}{\geq} \|\nabla h(x(t))\|_\star(\|\hat{f}(t) - f(t)\| + \|\hat{g}(t) - g(t)\|\|u(t)\|) \\ & \stackrel{(f)}{\geq} \|\nabla h(x(t))\|_\star(\|\hat{f}(t) - f(t)\| + \|(\hat{g}(t) - g(t))u(t)\|) \\ & \stackrel{(g)}{\geq} \langle \nabla h(x(t)), \hat{f}(t) + \hat{g}(t)u(t) \rangle - \langle \nabla h(x(t)), f(t) + g(t)u(t) \rangle \end{aligned}$$

where the inequality (e) follows since  $f(t) \in \mathcal{F}(x(t), t, )$  and  $g(t) \in \mathcal{G}(x(t), t, )$  due to Assumption 1 and where the inequality (f) follows by properties of the induced matrix norm  $\|\cdot\|$ . The inequality (g) follows again by application of Hölder's inequality. Consequently, by (15) it holds that  $\dot{h}(x(t)) \geq -\alpha(h(x(t)))$  for all  $t \in \mathcal{I}$ .

Next note that  $v(t) = -\alpha(v(t))$  with  $v(0) \geq 0$  admits a unique solution  $v(t)$  that is such that  $v(t) \geq 0$  for all  $t \geq 0$  [59, Lemma 4.4]. Using the Comparison Lemma [59, Lemma 3.4] and assuming

that  $h(x(0)) \geq 0$ , it follows that  $h(x(t)) \geq v(t) \geq 0$  for all  $t \in \mathcal{I}$ , i.e.,  $x(0) \in \mathcal{C}$  implies  $x(t) \in \mathcal{C}$  for all  $t \in \mathcal{I}$ . Recall next that (1) is defined on  $X(\mathcal{Y})$  and that  $\mathcal{Y} \supseteq Y(\mathcal{C})$  and hence  $X(\mathcal{Y}) \supseteq \mathcal{C}$  holds. Since  $x \in \mathcal{C}$  for all  $t \in \mathcal{I}$  and when  $\mathcal{C}$  is compact, it follows by [59, Theorem 3.3]<sup>6</sup> that  $\mathcal{I} = [0, \infty)$ , i.e.,  $\mathcal{C}$  is forward invariant under  $U(y, t)$ .

## B Proof of Proposition 1

Note first that, for any  $x \in \mathcal{N}$ , there exists a point  $x_i \in Z_{\mathcal{N}}$  satisfying  $\|x - x_i\| \leq \epsilon_{\mathcal{N}}$  since  $Z_{\mathcal{N}}$  is an  $\epsilon_{\mathcal{N}}$ -net of  $\mathcal{N}$ . For any  $x \in \mathcal{N}$ , we now select such an  $x_i \in Z_{\mathcal{N}}$  for which it follows that

$$\begin{aligned} h(x) &= h(x) - h(x_i) + h(x_i) \\ &\stackrel{(a)}{\leq} |h(x) - h(x_i)| - \gamma_{\text{unsafe}} \\ &\stackrel{(b)}{\leq} \text{Lip}_h(\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_i))\|x - x_i\| - \gamma_{\text{unsafe}} \\ &\stackrel{(c)}{\leq} \text{Lip}_h(\mathcal{B}_{\epsilon_{\mathcal{N}}}(x_i))\epsilon_{\mathcal{N}} - \gamma_{\text{unsafe}} \stackrel{(d)}{<} 0. \end{aligned}$$

Note that inequality (a) follows from constraint (7c), while inequality (b) follows by Lipschitz continuity. Inequality (c) follows by the assumption of  $Z_{\mathcal{N}}$  being an  $\epsilon_{\mathcal{N}}$ -net of  $\mathcal{N}$  and, finally, the strict inequality in (d) follows due to (8).

## C Proof of Proposition 2

The proof follows similarly to the proof of Proposition 1. For any  $x \in \underline{\mathcal{D}}$ , we select an  $x_i \in \underline{Z}_{\text{safe}}$  with  $\|x - x_i\| \leq \epsilon$  which is possible since the set  $\underline{Z}_{\text{safe}}$  is an  $\epsilon$ -net of  $\underline{\mathcal{D}}$ . It follows that

$$\begin{aligned} 0 &= h(x_i) - h(x_i) \\ &\stackrel{(a)}{\leq} h(x_i) - h(x) + h(x) - \gamma_{\text{safe}} \\ &\stackrel{(b)}{\leq} \text{Lip}_h(\mathcal{B}_{\epsilon}(x_i))\|x - x_i\| + h(x) - \gamma_{\text{safe}} \\ &\stackrel{(c)}{\leq} \text{Lip}_h(\mathcal{B}_{\epsilon}(x_i))\epsilon + h(x) - \gamma_{\text{safe}} \stackrel{(d)}{\leq} h(x). \end{aligned}$$

Note that inequality (a) follows from constraint (7b), while inequality (b) follows by Lipschitz continuity. Inequality (c) follows by  $\underline{Z}_{\text{safe}}$  being an  $\epsilon$ -net of  $\underline{\mathcal{D}}$  and, finally, the inequality in (d) follows due to (10).

## D Proof of Lemma 1

For each  $y \in \mathcal{Y}$ , there exists  $(u_i, y_i, t_i) \in Z_{\text{dyn}}$  such that

$$\|X(y) - \hat{X}(y_i)\| \leq \epsilon$$

---

<sup>6</sup>Note that the same result as in [59, Theorem 3.3] holds when the system dynamics are continuous, but not locally Lipschitz continuous.

by the definition of  $\mathcal{Y}$  as  $\mathcal{Y} = Y(\mathcal{D})$  and since the  $y$  components of  $Z_{\text{dyn}}$  transformed via  $\hat{X}$  form an  $\epsilon$ -net of  $\mathcal{D}$ . By Assumption 2, we also know that

$$\|X(y_i) - \hat{X}(y_i)\| \leq \bar{\Delta}_X$$

By Lipschitz continuity of  $Y$ , it hence follows that

$$\begin{aligned} \|y - y_i\| &= \|Y(X(y)) - Y(X(y_i))\| \\ &\leq \text{Lip}_Y(\bar{\mathcal{D}})\|X(y) - X(y_i)\| \\ &\leq \text{Lip}_Y(\bar{\mathcal{D}})(\|X(y) - \hat{X}(y_i)\| + \|\hat{X}(y_i) - X(y_i)\|) \\ &\leq \text{Lip}_Y(\bar{\mathcal{D}})(\epsilon + \bar{\Delta}_X) =: \bar{\epsilon}. \end{aligned}$$

Consequently, the  $y$  components of  $Z_{\text{dyn}}$  form an  $\bar{\epsilon}$ -net of  $\mathcal{Y}$ .

## E Proof of Proposition 3

Note first that, for each  $y \in \mathcal{Y}$ , there exists a pair  $(u_i, y_i, t_i) \in Z_{\text{dyn}}$  satisfying  $\|y - y_i\| \leq \bar{\epsilon}$  since the  $y$  component of  $Z_{\text{dyn}}$  form an  $\bar{\epsilon}$ -net of  $\mathcal{Y}$  by Lemma 1. For any pair  $(y, t) \in \mathcal{Y} \times \mathbb{R}_{\geq 0}$ , we now select such a pair  $(u_i, y_i, t_i) \in Z_{\text{dyn}}$  satisfying  $\|y - y_i\| \leq \bar{\epsilon}$  for which it follows that

$$\begin{aligned} 0 &\stackrel{(a)}{\leq} q(u_i, y_i, t_i) - \gamma_{\text{dyn}} \\ &\leq |q(u_i, y_i, t_i) - q(u_i, y, t_i)| + q(u_i, y, t_i) - \gamma_{\text{dyn}} \\ &\stackrel{(b)}{\leq} \text{Lip}_q(u_i, \mathcal{B}_{\bar{\epsilon}}(y_i), t_i)\|y_i - y\| + q(u_i, y, t_i) - \gamma_{\text{dyn}} \\ &\stackrel{(c)}{\leq} \text{Lip}_q(u_i, \mathcal{B}_{\bar{\epsilon}}(y_i), t_i)\bar{\epsilon} + q(u_i, y, t_i) - \gamma_{\text{dyn}} \\ &\leq \text{Lip}_q(u_i, \mathcal{B}_{\bar{\epsilon}}(y_i), t_i)\bar{\epsilon} + |q(u_i, y, t_i) - q(u_i, y, t)| + q(u_i, y, t) - \gamma_{\text{dyn}} \\ &\stackrel{(d)}{\leq} \text{Lip}_q(u_i, \mathcal{B}_{\bar{\epsilon}}(y_i), t_i)\bar{\epsilon} + \text{Bnd}_q(u_i, y_i) + q(u_i, y, t) - \gamma_{\text{dyn}} \\ &\stackrel{(e)}{\leq} q(u_i, y, t). \end{aligned}$$

Inequality (a) follows from constraint (7d). Inequality (b) follows by Lipschitz continuity, while inequality (c) follows since the  $y$  component of  $Z_{\text{dyn}}$  is an  $\bar{\epsilon}$ -net of  $\mathcal{Y}$ . Inequality (d) follows by the bound  $\text{Bnd}_q(u_i, y_i)$  that bounds the function  $q$  for all values of  $t$ . Inequality (e) follows simply by (12). Consequently,  $q(u_i, y, t) \geq 0$  for all  $(y, t) \in \mathcal{Y} \times \mathbb{R}_{\geq 0}$ .

Note next that  $\bar{\mathcal{D}} \supseteq \mathcal{X}(\mathcal{Y})$  for each  $y \in \mathcal{Y}$  so that, for each  $j \in \{1, 2, 3\}$ , we have that  $\text{Lip}_{B_j}(\bar{\mathcal{D}}, t) \geq \text{Lip}_{B_j}(\mathcal{X}(y), t)$  for each  $y \in \mathcal{Y}$ . If now additionally  $\bar{L}_j \geq \text{Lip}_{B_j}(\bar{\mathcal{D}}, t)$ , as stated per assumption, it follows immediately that (4) holds and that  $h(x)$  is a ROCBF.

## References

- [1] W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and decision-making for autonomous vehicles,” *An. Review Control, Robot., and Auton. Syst.*, vol. 1, pp. 187–210, 2018.

- [2] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, “Temporal-logic-based reactive mission and motion planning,” *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1370–1381, 2009.
- [3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Proc. Conf. Robot Learning*, Mountain View, California, November 2017, pp. 1–16.
- [4] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” in *Proc. Symp. Nonlin. Control Syst.*, Pretoria, South Africa, August 2007, pp. 462–467.
- [5] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *Proc. Conf. Decis. Control*, Los Angeles, CA, December 2014, pp. 6271–6278.
- [6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [7] P. Glotfelter, J. Cortés, and M. Egerstedt, “Nonsmooth barrier functions with applications to multi-robot systems,” *IEEE Control Syst. Lett.*, vol. 1, no. 2, pp. 310–315, 2017.
- [8] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multi-robot systems,” *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, 2017.
- [9] L. Lindemann and D. V. Dimarogonas, “Control barrier functions for signal temporal logic tasks,” *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 96–101, 2019.
- [10] W. Xiao and C. Belta, “Control barrier functions for systems with high relative degree,” in *Proc. Conf. Decis. Control*, Nice, France, December 2019, pp. 474–479.
- [11] S. Kolathaya and A. D. Ames, “Input-to-state safety with control barrier functions,” *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 108–113, 2018.
- [12] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” in *Proc. Conf. Analys. Design Hybrid Syst.*, Atlanta, GA, October 2015, pp. 54–61.
- [13] M. Jankovic, “Robust control barrier functions for constrained stabilization of nonlinear systems,” *Automatica*, vol. 96, pp. 359–367, 2018.
- [14] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, “Guaranteeing safety of learned perception modules via measurement-robust control barrier functions,” in *Proc. Conf. Robot Learning*, Boston, Massachusetts, November 2020, pp. 1–17.
- [15] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, “Measurement-robust control barrier functions: Certainty in safety with uncertainty in state,” *arXiv preprint arXiv:2104.14030*, 2021.
- [16] K. Garg and D. Panagou, “Robust control barrier and control lyapunov functions with fixed-time convergence guarantees,” in *Proc. Am. Control Conf.*, New Orleans, LA, May 2021, pp. 2292–2297.

- [17] P.-F. Massiani, S. Heim, F. Solowjow, and S. Trimpe, “Safe value functions,” *arXiv preprint arXiv:2105.12204*, 2021.
- [18] J. Ferlez, M. Elnaggar, Y. Shoukry, and C. Fleming, “ShieldNN: A provably safe nn filter for unsafe nn controllers,” *arXiv preprint arXiv:2006.09564*, 2020.
- [19] B. T. Lopez, J.-J. E. Slotine, and J. P. How, “Robust adaptive control barrier functions: An adaptive and data-driven approach to safety,” *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 1031–1036, 2020.
- [20] A. J. Taylor and A. D. Ames, “Adaptive safety with control barrier functions,” in *Proc. Am. Control Conf.*, Denver, CO, July 2020, pp. 1399–1405.
- [21] Y. Emam, P. Glotfelter, S. Wilson, G. Notomista, and M. Egerstedt, “Data-driven robust barrier functions for safe, long-term operation,” *arXiv preprint arXiv:2104.07592*, 2021.
- [22] A. Taylor, A. Singletary, Y. Yue, and A. Ames, “Learning for safety-critical control with control barrier functions,” in *Proc. Conf. Learning Dynamics Control*, San Francisco, CA, June 2020, pp. 708–717.
- [23] N. Csomay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, “Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety,” in *Proc. Conf. Learning Dynamics Control*, Zurich, Switzerland, June 2021, pp. 1041–1053.
- [24] H. Yin, P. Seiler, M. Jin, and M. Arcak, “Imitation learning with stability and safety guarantees,” *IEEE Control Syst. Lett.*, 2021.
- [25] 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. Conf. Artificial Intel.*, Honolulu, HI, February 2019, pp. 3387–3395.
- [26] L. Wang, E. A. Theodorou, and M. Egerstedt, “Safe learning of quadrotor dynamics using barrier certificates,” in *Proc. Conf. Robot. Automat.*, Brisbane, Australia, May 2018, pp. 2460–2465.
- [27] C. K. Verginis, F. Djemou, and U. Topcu, “Safety-constrained learning and control using scarce data and reciprocal barriers,” *arXiv preprint arXiv:2105.06526*, 2022.
- [28] E. D. Sontag, “A ‘universal’ construction of artstein’s theorem on nonlinear stabilization,” *Syst. & Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.
- [29] W. S. Cortez and D. V. Dimarogonas, “Correct-by-design control barrier functions for euler-lagrange systems with input constraints,” in *Proc. Am. Control Conf.*, Denver, CO, July 2020, pp. 950–955.
- [30] S. Prajna, A. Jadbabaie, and G. J. Pappas, “A framework for worst-case and stochastic safety verification using barrier certificates,” *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

- [31] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, “Correctness guarantees for the composition of lane keeping and adaptive cruise control,” *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1216–1229, 2017.
- [32] L. Wang, D. Han, and M. Egerstedt, “Permissive barrier certificates for safe stabilization using sum-of-squares,” in *Proc. Am. Control Conf.*, Milwaukee, WI, June 2018, pp. 585–590.
- [33] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *Proc. European Control Conf.*, Naples, Italy, June 2019, pp. 3420–3431.
- [34] A. Clark, “Verification and synthesis of control barrier functions,” *arXiv preprint arXiv:2104.14001*, 2021.
- [35] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, “Synthesis of control barrier functions using a supervised machine learning approach,” in *Proc. Conf. Intel. Robots Syst.*, Las Vegas, NV, October 2020, pp. 7139–7145.
- [36] M. Saveriano and D. Lee, “Learning barrier functions for constrained motion planning with dynamical systems,” in *Proc. Conf. Intel. Robot Syst.*, Macau, China, November 2019, pp. 112–119.
- [37] M. Ohnishi, G. Notomista, M. Sugiyama, and M. Egerstedt, “Constraint learning for control tasks with limited duration barrier functions,” *Automatica*, vol. 127, p. 109504, 2021.
- [38] S. Yaghoubi, G. Fainekos, and S. Sankaranarayanan, “Training neural network controllers using control barrier functions in the presence of disturbances,” in *Proc. Conf. Intel. Transp. Syst.*, Rhodes, Greece, September 2020, pp. 1–6.
- [39] W. Xiao, C. A. Belta, and C. G. Cassandras, “Feasibility-guided learning for constrained optimal control problems,” in *Proc. Conf. Decis. Control*, Jeju Islands, South Korea, December 2020, pp. 1896–1901.
- [40] S. M. Khansari-Zadeh and A. Billard, “Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions,” *Robot. Autonom. Syst.*, vol. 62, no. 6, pp. 752–765, 2014.
- [41] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, “Learning lyapunov functions for hybrid systems,” in *Proc. Conf. Hybrid Syst.: Comp. Control*, Nashville, TN, May 2021, pp. 1–11.
- [42] A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, and A. Peruffo, “FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks,” in *Proc. Conf. Hybrid Syst.: Comp. Control*, Nashville, TN, May 2021, pp. 1–11.
- [43] H. Dai, B. Landry, M. Pavone, and R. Tedrake, “Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems,” in *Conf. Decis. Control*, Jeju Islands, South Korea, December 2020, pp. 1274–1281.
- [44] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhwani, “Learning stability certificates from data,” in *Proc. Conf. Robot Learning*, Boston, Massachusetts, November 2020.

- [45] W. Jin, Z. Wang, Z. Yang, and S. Mou, “Neural certificates for safe control policies,” *arXiv preprint arXiv:2006.08465*, 2020.
- [46] H. Zhao, X. Zeng, T. Chen, Z. Liu, and J. Woodcock, “Learning safe neural network controllers with barrier certificates,” in *Proc. Symp. Depend. Software Eng.: Theories, Tools, Appl.*, Guangzhou, China, November 2020, pp. 177–185.
- [47] D. Sun, S. Jha, and C. Fan, “Learning certified control using contraction metric,” *arXiv preprint arXiv:2011.12569*, 2020.
- [48] K. Long, C. Qian, J. Cortés, and N. Atanasov, “Learning barrier functions with memory for robust safe navigation,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4931–4938, 2021.
- [49] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, “Learning safe multi-agent control with decentralized neural barrier certificates,” *arXiv preprint arXiv:2101.05436*, 2021.
- [50] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, “Learning control barrier functions from expert demonstrations,” in *Proc. Conf. Decis. Control*, Jeju Islands, South Korea, December 2020, pp. 3717–3724.
- [51] A. Robey, L. Lindemann, S. Tu, and N. Matni, “Learning robust hybrid control barrier functions for uncertain systems,” in *Proc. Conf. Anal. Design Hybrid Syst.*, Brussels, Belgium, July 2021, pp. 1–6.
- [52] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, “Learning hybrid control barrier functions from data,” in *Proc. Conf. Robot Learning*, Boston, Massachusetts, November 2020.
- [53] R. Vershynin, *High-dimensional prob.: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.
- [54] C. Xia, W. Hsu, M.-L. Lee, and B. C. Ooi, “Border: Efficient computation of boundary points,” *IEEE Trans. Knowledge Data Eng.*, vol. 18, no. 3, pp. 289–303, 2006.
- [55] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Proc. Advances Neur. Inform. Proc. Syst.*, Vancouver, Canada, December 2008, pp. 1177–1184.
- [56] A. Virmaux and K. Scaman, “Lipschitz regularity of deep neural networks: analysis and efficient estimation,” in *Proc. Advances Neur. Inform. Proc. Syst.*, Montréal, Canada, December 2018, pp. 3835–3844.
- [57] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, “Efficient and accurate estimation of lipschitz constants for deep neural networks,” in *Proc. Advances Neur. Inform. Proc. Syst.*, Vancouver, Canada, December 2019, pp. 11 423–11 434.
- [58] G. Wood and B. Zhang, “Estimation of the lipschitz constant of a function,” *Journ. of Global Opt.*, vol. 8, no. 1, pp. 91–103, 1996.
- [59] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.