In-Distribution Barrier Functions: Self-Supervised Policy Filters that Avoid Out-of-Distribution States

Fernando Castañeda FCASTANEDA@BERKELEY.EDU
University of California, Berkeley

Haruki Nishimura HARUKI.NISHIMURA@TRI.GLOBAL
Toyota Research Institute, Los Altos, CA

Rowan McAllister ROWAN.MCALLISTER@TRI.GLOBAL
Toyota Research Institute, Los Altos, CA

Koushil Sreenath KOUSHILS@BERKELEY.EDU
University of California, Berkeley

Adrien Gaidon ADRIEN.GAIDON@TRI.GLOBAL
Toyota Research Institute, Los Altos, CA

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

Learning-based control approaches have shown great promise in performing complex tasks directly from high-dimensional perception data for real robotic systems. Nonetheless, the learned controllers can behave unexpectedly if the trajectories of the system divert from the training data distribution, which can compromise safety. In this work, we propose a control filter that wraps any reference policy and effectively encourages the system to stay in-distribution with respect to offline-collected safe demonstrations. Our methodology is inspired by Control Barrier Functions (CBFs), which are model-based tools from the nonlinear control literature that can be used to construct minimally invasive safe policy filters. While existing methods based on CBFs require a known low-dimensional state representation, our proposed approach is directly applicable to systems that rely solely on high-dimensional visual observations by learning in a latent state-space. We demonstrate that our method is effective for two different visuomotor control tasks in simulation environments, including both top-down and egocentric view settings.

Keywords: Distributional Shift, Control Barrier Functions, State Representation Learning

1. Introduction

The modern advances in the representation learning literature have been an enabling factor for the recent surge of a wide variety of methods for robotic control directly from images or high-dimensional sensory observations (Watter et al., 2015; Ebert et al., 2018; Hafner et al., 2019; Lenz et al., 2015; Zhang et al., 2019; Van Hoof et al., 2016). These approaches for visuomotor planning and control have the potential to solve challenging tasks in which the state of the system might not be directly observable, or even not possible to model analytically. While promising, the high-dimensionality of the problem make these methods susceptible to several open challenges. For example, the exploration requirements of reinforcement learning (RL) algorithms are significantly exacerbated for these tasks, due to the high-dimensionality of the observations. This can result in abundant failures during training when trying to learn safe control policies. Supervised learning approaches for control, such as behavioral cloning, would in principle seem less prone to exhibit

unsafe behaviors. However, it is well known that simply because of their data-dependent nature, these methods are still susceptible to a key challenge named *distributional shift*: if the trajectories of the system divert from the training data distribution, the controller might take unexpected actions.

On the other hand, the control theory literature extensively covers the problem of long-horizon constraint satisfaction. In particular, Control Barrier Functions (CBFs, Ames et al. (2014)) are a popular model-based tool used to restrict the trajectories of the system from entering undesirable regions of the state-space. An attractive property of CBFs is that they decouple the problem of constraint-satisfaction from any performance objective. Indeed, if a CBF is available, then Ames et al. (2014) showed that we can construct a minimally invasive safety filter that transforms into safety-preserving control actions any unsafe commands that an arbitrary policy could output.

The main question we want to address in this work naturally emerges from the previous discussion: can we take inspiration from CBFs to avoid out-of-distribution (OOD) states when using data-driven controllers for visuomotor tasks? Even though CBFs are model-based tools that, as such, require knowledge of the state-space and dynamics of the system, the recent advances in learning latent state-space representations and associated dynamics models clearly set a path for linking data-driven visuomotor policy learning with the use of model-based control-theoretic tools such as CBFs.

Contributions: We present an end-to-end self-supervised approach for learning a task-agnostic policy filter which prevents the system from entering OOD states. We do not assume knowledge of the state-space or system dynamics. In addition, our framework only requires an offline-collected dataset of safe demonstrations (where the concept of safety is only linked to the demonstrator's subjectivity, as it is their responsibility to provide the dataset). We therefore do not require any unsafe demonstrations to learn a safe policy filter, in contrast to most other works tackling constrained policy learning. Furthermore, to the best of our knowledge, this is the first work that uses CBFs for constructing policy filters in learned latent state-spaces. This endows our approach with the flexibility of being applicable to systems with high-dimensional sensory observations, in contrast to most prior CBF-based methods. We present simulation experiments on two different visuomotor control tasks, which suggest that our framework, taking only raw RGB images as input, can learn to significantly reduce the distributional shift from safe demonstrations and, consequently, critically improve the safety of both systems.

2. Related Work

Several existing approaches for OOD prevention learn density models of training data that can be then used to restrict the agent from taking low likelihood actions or moving towards unvisited states (McAllister et al., 2019; Richter and Roy, 2017; Wu et al., 2019; Kumar et al., 2019). Although some of these methods have been shown to be effective at offline RL settings that are specially susceptible to distributional shift, the learned density models have no notion of control invariance and, therefore, do not consider the problem of how to prevent distributional shift over a long time horizon. A notable exception is the work of Kang et al. (2022) to constrain long-term distributional shift, in which a min-max Bellman backup operator is constructed so that Lyapunov-like functions arise as value functions of an offline RL problem. This work however does not consider the extension to visuomotor control tasks in learned latent spaces. Furthermore, for our approach we choose not to rely on a min-max backup operator to learn the certificate function and, instead, use the very suitable theory of CBFs to devise a self-supervised learning framework.

There exist some constructive procedures for synthesizing CBFs based on sum-of-squares programming (Jarvis-Wloszek et al., 2003; Majumdar et al., 2013; Dai and Permenter, 2022; Wang et al., 2022) or Hamilton-Jacobi reachability (Choi et al., 2021). However, these methods require knowledge of the dynamics of the system and typically suffer from scalability issues for high-dimensional systems. More in line with this work, some recent results show that CBFs can be learned from data (Jin et al., 2020; Dawson et al., 2022; Qin et al., 2022; Robey et al., 2020; Lindemann et al., 2021; Abate et al., 2021; Jagtap et al., 2020). None of these works, however, consider systems with high-dimensional observations. Furthermore, the works Jin et al. (2020); Dawson et al. (2022); Qin et al. (2022) assume a priori knowledge of a control-invariant safe set, and focus on building a CBF for that particular set. The line of research of Robey et al. (2020); Lindemann et al. (2021) has the most similar problem setup to our work, as they also consider learning from safe demonstrations. Although, notably, the authors provide formal verification arguments for the learned CBFs, their methods are not applicable to high-dimensional observations, assume a nominal dynamics model is given, and use an algorithmic approach for the detection of the boundary of the dataset that does not scale to large datasets. Other recent approaches build signed distance functions from sensory measurements that are obtained from a LiDAR or stereo cameras (Long et al., 2021; Srinivasan et al., 2020; Cosner et al., 2022). However, these functions are not encouraged to satisfy any set invariance property.

Extensions of the CBF-based control filters to systems with dynamics or measurement-model uncertainty have also been recently proposed (Nguyen and Sreenath, 2021; Castañeda et al., 2021; Taylor et al., 2021; Dhiman et al., 2021; Dean et al., 2021). These works assume that a CBF is provided, and formulate uncertainty-robust optimization problems for the controller design. They can be considered complementary to our deterministic but end-to-end approach. Future work should explore quantifying uncertainty estimates within our framework to robustify the learned policy filters.

Finally, the work of Wilcox et al. (2022) presents a framework to learn safe sets in a latent state-space for iterative control tasks. Compared to this work, our framework has the advantage that it is task-agnostic and does not require any interactions with the environment during training.

3. Background on Control Barrier Functions

We start by introducing some necessary background on Control Barrier Functions, which are tools from the nonlinear control literature that serve to enforce safety constraints for systems with known dynamics. As will be clear later, CBFs are particularly well-suited for continuous-time nonlinear control-affine systems of the form

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ is the state and $u \in \mathcal{U} \subset \mathbb{R}^m$ the control input. We assume that $f : \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{X} \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous.

In the CBF literature, safety is considered as a set invariance problem. In particular, we say that a control policy $\pi : \mathcal{X} \to \mathcal{U}$ assures the safety of system (1) with respect to a set $\mathcal{X}_{\text{safe}} \subset \mathcal{X}$ if the set $\mathcal{X}_{\text{safe}}$ is forward invariant under the control law π, i.e., for any $x_0 \in \mathcal{X}_{\text{safe}}$, the solution $x(t)$ of system (1) under the control law π remains within $\mathcal{X}_{\text{safe}}$ for all $t \geq 0$.

Definition 1 (Control Barrier Function, Ames et al. (2017)) *We say that a continuously differentiable function $B : \mathcal{X} \to \mathbb{R}$ is a* Control Barrier Function (CBF) *for system (1) with associated safe-set $\mathcal{X}_{safe} \subset \mathcal{X}$ if the following three conditions are satisfied:*

$$B(x) \geq 0 \quad \forall x \in \mathcal{X}_{safe}, \tag{2}$$

$$B(x) < 0 \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_{safe}, \tag{3}$$

$$\exists u \in \mathcal{U} \;\; s.t. \;\; \dot{B}(x,u) + \gamma(B(x)) \geq 0 \quad \forall x \in \mathcal{X}, \tag{4}$$

where $\gamma : \mathbb{R} \to \mathbb{R}$ is an extended class $\mathcal{K}_\infty$ function.

The existence of a CBF B guarantees that for system (1) any Lipschitz continuous control policy π satisfying

$$\pi(x) \in \{u \in \mathcal{U} : \underbrace{\nabla B(x)[f(x) + g(x)u]}_{=\dot{B}(x,u)} + \gamma\left(B(x)\right) \geq 0\} \tag{5}$$

will render the set $\mathcal{X}_{\text{safe}}$ forward invariant (Ames et al., 2017, Corollary 2).

For a given task-specific reference controller $\pi_{\text{ref}} : \mathcal{X} \to \mathcal{U}$ that might be safety-agnostic, the condition of (5) can be used to formulate an optimization problem that, when solved at every time-step, yields a minimally-invasive policy safety filter (Ames et al., 2014):

$$\pi_{\text{CBF}}(x) = \underset{u \in \mathcal{U}}{\arg\min} \quad \|u - \pi_{\text{ref}}(x)\|^2 \qquad \text{(CBF-QP)}$$
$$\text{s.t. } \nabla B(x)[f(x) + g(x)u] + \gamma(B(x)) \geq 0.$$

Assuming that the actuation constraints that define $\mathcal{U}$ are linear in u, this problem is a quadratic program (QP). This is a consequence of the dynamics of the system (1) being control-affine, and it practically means that the problem can be solved to a high precision very quickly (around 10^3Hz). This is critical since the CBF-QP needs to be solved at the real-time control frequency.

4. Problem Statement

The CBF-QP constitutes a very appealing approach for practitioners: it provides a task-agnostic minimally invasive filter that can wrap safety around any given policy π_{ref}, and therefore rewrite any unsafe control input that π_{ref} could output at any time. However, designing a valid CBF is nontrivial. In fact, it is still an active research topic even when assuming perfect knowledge of the dynamics of the system (Dai and Permenter, 2022; Choi et al., 2021; Wang et al., 2022). The two main difficulties in the design of a CBF are the following: first, a control-invariant set $\mathcal{X}_{\text{safe}}$ must be obtained (which in general is different from the geometric constraint set that could be obtained, for instance, from a signed-distance field) and, second, a function that satisfies condition (4) must be found for that set. Furthermore, even after obtaining a CBF, solving the CBF-QP requires perfect state and dynamics knowledge.

With our framework, we take initial steps towards building a safe policy filter from high-dimensional observations. Specifically, we take inspiration from CBFs to design an end-to-end learning framework to constrain deep learning models to remain in-distribution of the training data. We take as input a dataset of high-dimensional observations of different safe demonstrations, and build a neural CBF-like function that encourages the system to always stay in-distribution with respect to the observations from the safe demonstrations. This, in turn, significantly improves the safety of the system during deployment.

More concretely, for a given dataset of N safe trajectories $\mathbb{D} = \left\{ \left(I_t^i, u_t^i\right)_{t=0}^{t=T_i} \right\}_{i=1}^{i=N}$ sampled from the data-generating distribution of the demonstrator, we tackle the problem of designing a policy filter that can be applied to any reference controller π_{ref} to detect and override actions from

π_{ref} that lead to OOD states. We denote I_t^i and u_t^i the high-dimensional observation and control input, respectively, measured at time t for the ith trajectory. Furthermore, T_i is the final time-step of trajectory i. The demonstrations in the dataset $\mathbb{D}$ might correspond to different tasks and they do not need to be optimal with respect to any objective. In fact, our only assumption is that the dataset only contains safe demonstrations (in the sense that these trajectories should not contain any states from which the system is deemed to fail, even if it has not failed yet), so that we can encourage long-term constraint satisfaction using CBFs.

5. In-Distribution Barrier Functions

In this section, we introduce a self-supervised approach for synthesizing neural CBF-like functions whose aim is to constrain the system to remain in-distribution with respect to an offline dataset of safe demonstrations. We call these functions *in-Distribution Barrier Functions* (iDBFs). We will for now assume that we have a parametric continuous-time control-affine model of the dynamics of the system in a state-space $\mathcal{X} \subset \mathbb{R}^n$

$$\dot{x} = f_\theta(x) + g_\theta(x)u, \tag{6}$$

and present the iDBF learning procedure for this system. Furthermore, for this section we assume that the dataset $\mathbb{D}$ of safe trajectories contains true state measurements, i.e., $\mathbb{D} = \left\{ \left(x_t^i, u_t^i\right)_{t=0}^{t=T_i} \right\}_{i=1}^{i=N}$, where x_t^i and u_t^i are the state and control input, respectively, measured at time t for the ith trajectory. In Section 6, we will provide details on how to learn a dynamics model of this form in a latent state-space when we have a dataset containing high-dimensional sensory observations.

We parameterize an iDBF $B_\phi : \mathcal{X} \to \mathbb{R}$ as a neural network with parameters ϕ, and construct an empirical loss function that encourages it to satisfy the three CBF conditions (2), (3) and (4) with respect to a set $\mathcal{X}_{\text{safe}}$ that is also implicitly learned through self-supervision. To design the loss function, we take inspiration from previous literature on learning CBFs (Dawson et al., 2022; Qin et al., 2022; Chang et al., 2019). Nevertheless, instead of assuming that the safe-set $\mathcal{X}_{\text{safe}}$ is given and that we can sample from it and from its unsafe complement $\mathcal{X}_{\text{unsafe}} \doteq \mathcal{X} \setminus \mathcal{X}_{\text{safe}}$, we build our loss function in a self-supervised manner just from the dataset of safe demonstrations. We accomplish this by leveraging ideas from contrastive learning (Gutmann and Hyvärinen, 2010; Oord et al., 2018; Chopra et al., 2005; Weinberger and Saul, 2009; Schroff et al., 2015). In particular, as we explain in detail later, we build a contrastive distribution from which to sample candidate unsafe states, given that we do not have any unsafe demonstrations in our dataset. The loss function we propose for learning an iDBF takes the following form:

$$\begin{aligned}\mathcal{L}_{\text{iDBF}} = \frac{w_{\text{safe}}}{N_{\text{safe}}} \sum_{x_{\text{safe}}} \left[\epsilon_{\text{safe}} - B_\phi(x_{\text{safe}})\right]^+ + \frac{w_{\text{unsafe}}}{N_{\text{unsafe}}} \sum_{x_{\text{unsafe}}} \left[\epsilon_{\text{unsafe}} + B_\phi(x_{\text{unsafe}})\right]^+ + \\ \frac{w_{\text{ascent}}}{N_{\text{safe}}} \sum_{(x_{\text{safe}}, u_{\text{safe}})} \left[\epsilon_{\text{ascent}} - \left(\nabla B_\phi(x_{\text{safe}})[f_\theta(x_{\text{safe}}) + g_\theta(x_{\text{safe}})u_{\text{safe}}] + \gamma(B_\phi(x_{\text{safe}}))\right)\right]^+,\end{aligned} \tag{7}$$

where $[\cdot]^+ \coloneqq \max(0, \cdot)$; $(x_{\text{safe}}, u_{\text{safe}})$ are a batch of N_{safe} samples from the empirical distribution of the dataset $\mathbb{D}$; x_{unsafe} are N_{unsafe} samples from a contrastive distribution that we will define soon; w_{safe}, w_{unsafe} and w_{ascent} are the weights of each loss term; and ϵ_{safe}, ϵ_{unsafe} and ϵ_{ascent} are positive constants that serve to enforce strictly the inequalities and generalize outside of the training data.

The goal of the first two terms in the loss function is to learn an iDBF that has a positive value in states that belong to the data distribution of safe demonstrations, and negative everywhere else (aligning with conditions (2) and (3) of the definition of CBF). Note that this classification objective

is very related to the notion of energy-based models (EBMs)—neural density models that assign a low energy value to points close to the training data distribution and a high value to distant ones (Hinton, 2002). In fact, we took inspiration from the Noise Contrastive Estimation (NCE, Gutmann and Hyvärinen (2010)) training procedure of EBMs, in particular the InfoNCE loss (Oord et al., 2018), to design (7). Intuitively, these methods use a noise contrastive distribution to generate candidate examples where to increase the value of the energy of the EBM, while decreasing the energy at the training data points. We precisely aim for the reverse result: a high value of B_ϕ on the training data distribution, and a low value everywhere else. However, we have one additional requirement, which is that the iDBF should have a value of zero at the boundary, as set by conditions (2) and (3). Hence, we design the two first terms of the loss function using the $[\cdot]^+$ operator.

In the third term of the loss (7), note that we do not encourage the satisfaction of condition (4) over the entire state-space, but only over the dataset of safe demonstrations. However, even if condition (4) is only satisfied $\forall x \in \mathcal{X}_{\text{safe}}$ instead of $\forall x \in \mathcal{X}$, a CBF still guarantees the control-invariance of $\mathcal{X}_{\text{safe}}$. Thus, we use our empirical data distribution of safe demonstrations as a sampling distribution covering the set $\mathcal{X}_{\text{safe}}$, which we are also implicitly learning as the zero-superlevel set of B_ϕ. Furthermore, unlike prior approaches that encourage the satisfaction of condition (4) for a single policy (Dawson et al., 2022; Qin et al., 2022), we instead use all pairs $(x_{\text{safe}}, u_{\text{safe}})$ present in the dataset $\mathbb{D}$ to compute this term of the loss. This way, we force the set of admissible control inputs (5) to be as large as our dataset allows, reducing the conservatism of the learned iDBF.

In order to generate the contrastive distribution from which to sample x_{unsafe}, as we ultimately want to learn the iDBF in a latent state-space in which it might not be intuitive how to construct a noise distribution, we take the following steps. 1) Based on the dataset of safe demonstrations $\mathbb{D}$, we train a neural behavioral cloning (BC) model that outputs a multi-modal Gaussian distribution over actions conditioned on the state, with density $\pi_{\text{BC}}(u|x)$. 2) Then, during the training process of the iDBF, for each x_{safe} state sampled from $\mathbb{D}$ we randomly take $N_{\text{candidate}}$ control inputs $u_{\text{candidate}}$ and evaluate their density value based on the BC model $\pi_{\text{BC}}(u_{\text{candidate}}|x_{\text{safe}})$. 3) If the value of the density falls below a threshold, then that control input is forward-propagated for one timestep using the dynamics model (6) to generate a sample x_{unsafe}. This way, we generate a contrastive data distribution by propagating actions that are unlikely present in the dataset of safe demonstrations. Furthermore, by only propagating these actions for one timestep, the contrastive distribution is close to the training data, which is desirable for the learning process (Gutmann and Hirayama, 2011).

This contrastive learning approach to distinguish between safe and unsafe states scales to large datasets, unlike algorithmic boundary-detection methods as in Lindemann et al. (2021).

6. Learning iDBFs from High-Dimensional Observations

After introducing the training procedure for an iDBF when the state representation and dynamics model (6) are given, we now relax these assumptions and present an approach to learn a latent state-space representation and a continuous-time dynamics model of the form (6), suitable to be integrated in the same end-to-end learning framework. We therefore now consider precisely the problem setting described in Section 4, in which we only assume having access to a dataset containing observation-action pairs of safe demonstrations $\mathbb{D} = \left\{ \left(I_t^i, u_t^i \right)_{t=0}^{t=T_i} \right\}_{i=1}^{i=N}$.

We use an autoencoder architecture to obtain the latent state-space representation, and employ the training procedure of Neural Ordinary Differential Equations (Neural ODEs, Chen et al. (2018)) to learn a dynamics model of the form (6) in the latent state-space. Note that by enforcing the

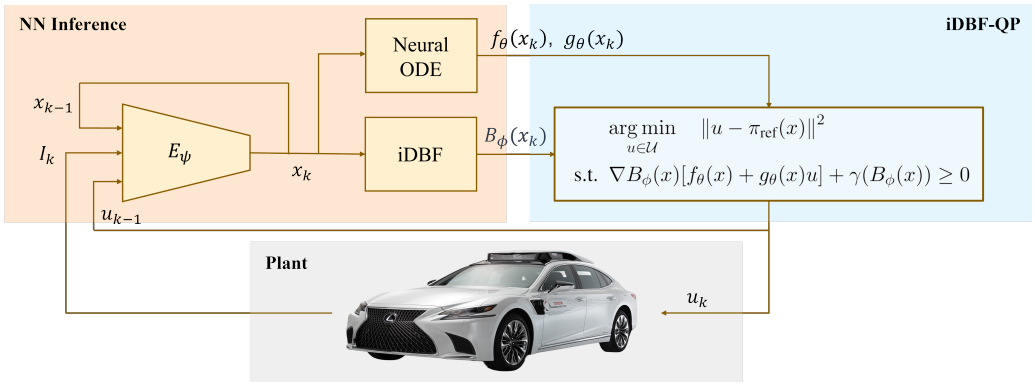

Figure 1: Our framework's inference diagram. At each timestep, based on the current observation I_k and the previous latent state x_{k-1} and action u_{k-1}, the encoder network E_ψ outputs a new latent state x_k. Then, the iDBF and dynamics networks give the values of $B_\phi(x_k)$, $f_\theta(x_k)$ and $g_\theta(x_k)$ which are passed on to the iDBF-QP policy filter. The iDBF-QP takes a reference control input for the current timestep $\pi_{\text{ref}}(x_k)$ and returns the closest action that keeps the system in-distribution with respect to the offline-collected dataset of safe demonstrations. For both of the examples of Section 7, the total inference time (NN Inference + solving the iDBF-QP) of our framework is less than 5 milliseconds.

continuous-time control-affine structure of the dynamics model, we ensure that the iDBF-QP policy filter (equivalent to the CBF-QP, see Figure 1) obtained with the learned iDBF and dynamics model will also be a quadratic program.

The inference procedure of our end-to-end learning framework is depicted in Figure 1. We use a recursive encoder network E_ψ that takes the current measurement I_k, as well as the previous latent state x_{k-1} and action u_{k-1} to generate the new latent state x_k at each time-step k. The decoder network D_ξ generates a reconstructed observation $\hat{I}_k$ for each latent state x_k. The proposed loss function for the latent state-space representation and dynamics model penalizes both the prediction error of the dynamics model and the observation reconstruction error:

$$\mathcal{L}_{\text{dyn}} = \frac{1}{N_{\text{dyn}}(T_{\text{pred}}+1)} \sum_{j=1}^{N_{\text{dyn}}} \sum_{k=0}^{T_{\text{pred}}} \left[w_{\text{state}} \left\| \tilde{x}_{t_j+k|t_j} - x_{t_j+k} \right\|^2 + w_{\text{rec}_1} \left\| \tilde{I}_{t_j+k|t_j} - I_{t_j+k} \right\|^2 + w_{\text{rec}_2} \left\| \hat{I}_{t_j+k} - I_{t_j+k} \right\|^2 \right]. \quad (8)$$

Here, $x_{t_j+k} = E_\psi(I_{t_j+k}, x_{t_j+k-1}, u_{t_j+k-1})$ is the latent state at timestep t_j+k. $\tilde{x}_{t_j+k|t_j}$ denotes the latent state prediction obtained by forward-propagating the dynamics model (6) to timestep t_j+k starting from the state x_{t_j} and using zero-order hold on the sequence of control inputs $(u_{t_j}, u_{t_j+1}, ..., u_{t_j+k-1})$. Additionally, $\tilde{I}_{t_j+k|t_j} := D_\xi(\tilde{x}_{t_j+k|t_j})$ is the reconstructed observation from the dynamics prediction for timestep t_j+k. Finally, $\hat{I}_{t_j+k} := D_\xi(x_{t_j+k})$ is the encoded-decoded observation at timestep t_j+k.

Note that we use a multiple-shooting error for the loss (8), as the prediction horizon T_{pred} does not need to coincide with the length of the trajectories in the dataset $\mathbb{D}$. In particular, the loss (8) is computed by sampling a batch of trajectories from $\mathbb{D}$ and then splitting them into N_{dyn} portions of length T_{pred}. The initial timestep of each portion $j = 1, ..., N_{\text{dyn}}$ is denoted as t_j. The first two terms in the loss function are then penalizing the state and reconstruction error of the multistep predictions of the dynamics model from each initial state x_{t_j}. The last term in the loss function penalizes the reconstruction error of the autoencoder directly, without using the dynamics model. The recent results of Beintema et al. (2021) show that multiple shooting loss functions lead to more accurate predictions compared to single-step prediction losses, and to better conditioned learning problems compared to single-shooting propagation losses.

An iDBF can be learned together with the autoencoder and dynamics model by optimizing jointly the losses (7) and (8). For the iDBF loss, each x_{safe} is obtained by encoding the observations

		π_{ref}	Ours	BC Filter			Ensemble Filter		
				p_{low}	p_{mid}	p_{high}	δ_{low}	δ_{mid}	δ_{high}
Top-Down Navigation	Collision Rate (%)	$\mathbf{46.72 \pm 8.36}$	$\mathbf{0.28 \pm 0.27}$	$\mathbf{35.60 \pm 7.20}$	$\mathbf{13.86 \pm 4.96}$	$\mathbf{2.48 \pm 1.57}$	$\mathbf{43.92 \pm 7.90}$	$\mathbf{43.82 \pm 7.41}$	$\mathbf{42.88 \pm 7.41}$
	Cumulative Intervention	0.0 ± 0.0	109.2 ± 20.1	85.6 ± 6.8	146.4 ± 9.6	189.1 ± 11.6	150.2 ± 19.3	94.5 ± 16.3	52.7 ± 11.5
Egocentric Driving	Collision Rate (%)	$\mathbf{81.00 \pm 0.23}$	$\mathbf{1.56 \pm 1.20}$	$\mathbf{21.94 \pm 1.85}$	$\mathbf{14.44 \pm 2.69}$	$\mathbf{8.78 \pm 1.86}$	$\mathbf{78.74 \pm 0.20}$	$\mathbf{78.60 \pm 1.63}$	$\mathbf{81.50 \pm 0.27}$
	Cumulative Intervention	0.0 ± 0.0	278.1 ± 32.6	713.8 ± 1.4	726.7 ± 2.6	750.8 ± 6.9	28.7 ± 2.1	42.8 ± 3.6	208.9 ± 5.7

Table 1: Evaluation of the collision rate and cumulative filter intervention (a measure of how intrusive the filter is with respect to the reference controller) for the top-down view robotic navigation example (over 20 simulations of 5-seconds each with random initial and goal states) and for the egocentric view autonomous driving example (over 20 simulations of 50-seconds each with random initial heading angles). For the BC and ensemble filters, we provide results for 3 different threshold values: $(p_{\text{low}}, p_{\text{mid}}, p_{\text{high}}) = (0.32, 0.35, 0.38)$ for the navigation example, and $(0.2, 0.5, 0.8)$ for driving; and $(\delta_{\text{low}}, \delta_{\text{mid}}, \delta_{\text{high}}) = (0.0005, 0.001, 0.002)$ for both examples.

sampled from the dataset $\mathbb{D}$, and x_{unsafe} is obtained by forward propagating the actions that have a low probability according to the pretrained BC model, as explained at the end of last section.

Once the iDBF B_ϕ; dynamics model f_θ and g_θ; and encoder E_ψ networks are trained, we can construct a policy filter —which we call iDBF-QP in Figure 1— in an equivalent manner to the CBF-QP that was introduced in Section 3.

Remark 2 *It is important to note that our iDBF training procedure encourages the satisfaction of the CBF conditions* (2), (3) *and* (4) *only at a discrete set of training points (which has measure zero). Because of this, we do not have control invariance guarantees for any particular set, and solving the iDBF-QP does not theoretically assure that the system will remain in-distribution. Although obtaining rigorous theoretical guarantees should be a priority for future work, the empirical results of Section 7 show that our framework takes a promising first step towards building effective policy filters from raw high-dimensional observations.*

7. Examples

In this section, we present the empirical evaluation of our framework on two different simulation environments: a toy example of a robot navigation task using top-down images of the scene, and an autonomous driving scenario with egocentric image observations. For both cases, given a safety-agnostic reference controller π_{ref}, we use our iDBF-QP at each timestep with the latest image measurement to find the closest control input to π_{ref} among those that prevent the system from entering OOD states (see Figure 1). For each environment, we train the iDBF, autoencoder and dynamics model using a dataset containing 64×64 RGB images of offline-collected trajectories.

Robot Navigation with Top-Down View Images: In this example, a circular robot with radius of 1 meter navigates inside of a 10×10 meter room that has a square-shaped 4×4 meter static obstacle in the middle, as shown in Figure 2 (left). The underlying dynamics of the robot are those of a 2D single integrator, with two control inputs corresponding to the x and y velocity commands, although we do not assume having access to that knowledge. Instead, we only have a dataset of image-action pairs corresponding to 5000 trajectories of 100 points each (corresponding to 2 seconds since the time-step is 0.02s). These trajectories satisfy two requirements: 1) the robot should never collide against the obstacle, and 2) the center of the robot should never leave the room limits. The trajectories are collected applying random actions at each time-step, and we check both conditions before adding a trajectory to the dataset. We use our framework to train an autoencoder with latent state-space of dimension 3, a dynamics model, and an iDBF. The reference policy π_{ref} simply

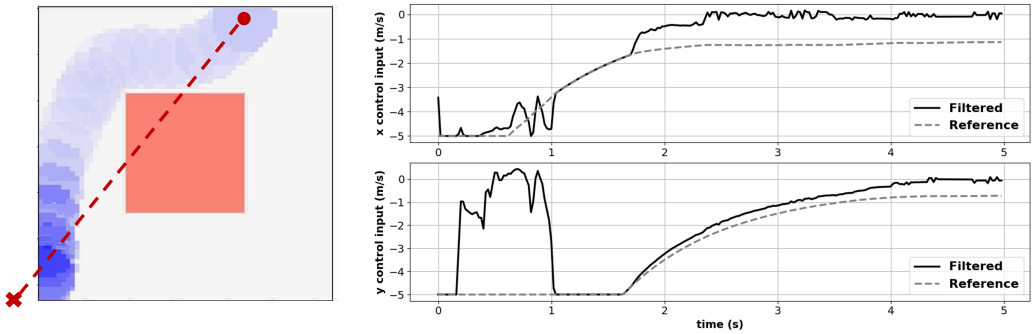

Figure 2: Example result using our proposed policy filter for a robot top-down visual navigation task. The reference controller simply tries to bring the robot (blue circle) to a goal state (denoted with $\times$). Our proposed filter, by keeping the system in-distribution, prevents the robot from colliding against the obstacle (orange square) and keeps its center-point inside the limits of the image. A video with several demonstrations of our approach for this task can be found in this link.

applies a velocity in the direction of a goal-point, with magnitude proportional to the distance. In Figure 2, we show the results of applying our iDBF-QP when the goal state (marked with an $\times$) is outside of the room limits and at the other side of the obstacle. Even though the reference controller is trying to take the shortest path, which would go through the obstacle, the iDBF-QP prevents the robot from first, colliding with the obstacle, and second, from having its center exit the room limits.

Autonomous Driving with Egocentric View Images: We use the environment provided by Kahn et al. (2018), which is based on the Bullet physics simulator and the Panda3d graphics engine (Goslin and Mine, 2004) to obtain egocentric RGB image measurements. The car navigates in a corridor which has four 90-degree turns to form a square-shaped center-line. One of such turns is shown in the snapshots of Figure 3. The car has two control inputs: the desired forward velocity and the steering angle. Given the high-order dynamics of the simulator, we collect data manually to make sure no trajectories included in the dataset are deemed to collide with any of the walls. We split the collected data into 450 trajectories of 100 points each (5 seconds since the timestep is 0.05s). This makes for a much sparser and less diverse (since it is collected by a human) dataset compared to the previous example. During deployment, we use a reference controller π_{ref} that simply drives the car forward at a constant speed of $3.5m/s$. Our iDBF-QP framework of Figure 1, taking the latest egocentric RGB measurement as input and using a latent state-space of dimension 3, is very effective at preventing the car from colliding against the walls, as shown in Table 1. Figure 3 contains snapshots of how our iDBF-QP forces the car to turn as it approaches a corner, even though the reference command is to drive forward.

Using these simulation environments we also aim to compare our proposed approach with other techniques for avoiding distributional shift. Other works that consider this problem use data density models to constrain the learned policies (Richter and Roy, 2017; McAllister et al., 2019; Wu et al., 2019), or use uncertainty estimation schemes, such as ensemble models, to avoid taking actions that lead to highly uncertain states (Chua et al., 2018). We build our baselines upon a conditional BC density model of the training data and an ensemble of latent state-space dynamics models:

BC Density Filter Baseline: As explained in Section 5, we train a BC multi-modal Gaussian model that is used to generate the contrastive training distribution for the iDBF. For any state, the BC model outputs a probability distribution over actions, with density function $\pi_{\text{BC}}(u|x)$. We train this BC model using privileged true-state information of the system, and use its density values to build a filter that serves as an apples-to-apples baseline comparison to our approach. Specifically, the baseline also takes the reference controller π_{ref} and, at every timestep, it finds the closest control action to $\pi_{\text{ref}}(x)$ that satisfies $\pi_{\text{BC}}(u|x) \geq p$, out of 200 randomly sampled actions. Note that this

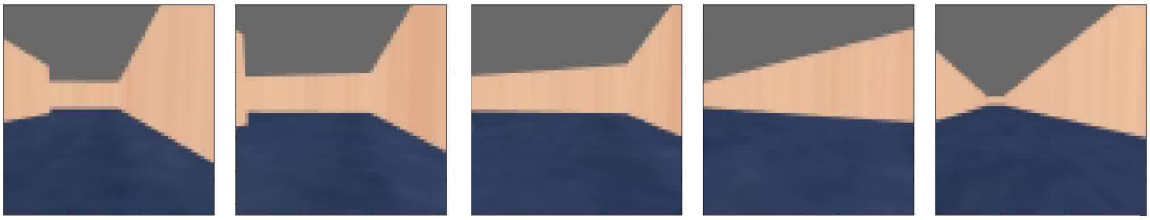

Figure 3: Snapshots of egocentric view images of a driving simulation when the car is approaching a corner. The reference controller just commands the car to drive straight, but our iDBF-QP policy filter forces a left turn as the car approaches the corner. Therefore, our filter prevents a collision as a result of staying in-distribution with respect to the safe training data. A video with several demonstrations of our approach for this task can be found in this link.

means that the baseline has access to true-state information at test-time, while our framework lacks this advantage. If no control action satisfying that condition is found, the reference control input is applied without filtering. Given the baseline's dependence on the threshold value p, we test multiple p values and present the outcomes for three representative cases $p_{\text{low}},\ p_{\text{mid}},\ p_{\text{high}}$ in Table 1.

Ensemble Variance Filter Baseline: We also train an ensemble of independent latent state-space dynamic models (f_θ and g_θ), keeping the rest of the framework introduced in Section 6 unchanged. During deployment, at every timestep we look for the closest control action to $\pi_{\text{ref}}(x)$ that keeps the variance $\sigma^2_{\text{ens}}(x,u)$ of the predicted dynamics $f_\theta(x) + g_\theta(x)u$ under a threshold δ. As in the previous baseline, we also look over 200 randomly sampled actions at each timestep, and different threshold levels $\delta_{\text{low}},\ \delta_{\text{mid}}$ and δ_{high}. Again, if no control action satisfying the threshold condition is found, the reference control input is applied without filtering.

In Table 1, we provide a summary of the comparison results for both environments. We use the collision rate as a proxy for distributional shift, since the training data only includes collision-free trajectories. The collision rate for the robot navigation example is computed as the fraction of time that the robot spends either in collision with the obstacle or having its center-point outside of the room limits. For the driving scenario, the collision rate is the fraction of time that the robot is in collision with any of the walls. For both examples, our method drastically reduces the collision rate compared to using the reference (unfiltered) controller. Furthermore, we achieve the lowest collision rates when compared to the baselines. From the baselines, only the BC density filter (with a very restrictive threshold p_{high}) manages to achieve small collision rates, at the cost of a very high cumulative filter intervention rate. The filter intervention rate is computed for both examples as $\sum_t \|u_t - \pi_{\text{ref}}(x_t)\|^2$, where each control input dimension is normalized between -1 and 1.

8. Conclusion

In this work, we take first-steps towards merging control-theoretic CBFs with practical robotic tasks that involve high-dimensional perception modules. We consider a realistic problem setting in which no unsafe demonstrations are available, and take a self-supervised learning approach to learn a function that effectively restricts the system from diverging towards OOD states. By learning this function in a latent state-space, our framework should be flexible-enough to be applicable to a wide variety of visuomotor tasks, and should be compatible with the use of large-scale pretrained representation learning models. Another important direction for future work would be to use probabilistic encoding and dynamics models to be able to robustify our proposed filters with respect to prediction uncertainties. Additionally, exploring the use of loss functions that are not based on reconstruction, by exploiting the value function nature of the iDBF, could be another promising direction.

Acknowledgments

The authors would like to thank Dr. Jean Mercat, Dr. Hongkai Dai, Dr. Katherine Liu and Blake Wulfe for their insightful comments and suggestions.

References

Alessandro Abate, Daniele Ahmed, Alec Edwards, Mirco Giacobbe, and Andrea Peruffo. Fossil: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2021.

A. D. Ames, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *IEEE Conference on Decision and Control*, pages 6271–6278, 2014.

A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62:3861–3876, 2017.

Gerben Beintema, Roland Toth, and Maarten Schoukens. Nonlinear state-space identification using deep encoder networks. In *Learning for Dynamics and Control*, pages 241–250. PMLR, 2021.

Fernando Castañeda, Jason J. Choi, Bike Zhang, Claire J. Tomlin, and Koushil Sreenath. Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty. In *IEEE Conference on Decision and Control*, pages 6762–6769, 2021.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Jason J Choi, Donggun Lee, Koushil Sreenath, Claire J Tomlin, and Sylvia L Herbert. Robust control barrier–value functions for safety-critical control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6814–6821. IEEE, 2021.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Ryan K Cosner, Ivan D Jimenez Rodriguez, Tamas G Molnar, Wyatt Ubellacker, Yisong Yue, Aaron D Ames, and Katherine L Bouman. Self-supervised online learning for safety-critical control using stereo vision. *arXiv preprint arXiv:2203.01404*, 2022.

Hongkai Dai and Frank Permenter. Convex synthesis and verification of control-lyapunov and barrier functions with input constraints. *arXiv preprint arXiv:2210.00629*, 2022.

Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, pages 1724–1735. PMLR, 2022.

Sarah Dean, Andrew Taylor, Ryan Cosner, Benjamin Recht, and Aaron Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning*, 2021.

Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, 2021.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Mike Goslin and Mark R Mine. The panda3d graphics engine. *Computer*, 37(10):112–114, 2004.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

Michael U Gutmann and Jun-ichiro Hirayama. Bregman divergence as general framework to estimate unnormalized statistical models. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 283–290, 2011.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Pushpak Jagtap, George J Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using gaussian processes. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3699–3704. IEEE, 2020.

Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard. Some controls applications of sum of squares programming. In *42nd IEEE international conference on decision and control*, volume 5, pages 4676–4681. IEEE, 2003.

Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. Neural certificates for safe control policies. *arXiv preprint arXiv:2006.08465*, 2020.

Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, and Sergey Levine. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5129–5136. IEEE, 2018.

Katie Kang, Paula Gradu, Jason J Choi, Michael Janner, Claire Tomlin, and Sergey Levine. Lyapunov density models: Constraining distribution shift in learning-based control. In *International Conference on Machine Learning*, pages 10708–10733. PMLR, 2022.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Ian Lenz, Ross A Knepper, and Ashutosh Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, volume 10. Rome, Italy, 2015.

Lars Lindemann, Alexander Robey, Lejun Jiang, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. *arXiv preprint arXiv:2111.09971*, 2021.

Kehan Long, Cheng Qian, Jorge Cortés, and Nikolay Atanasov. Learning barrier functions with memory for robust safe navigation. *IEEE Robotics and Automation Letters*, 6(3):4931–4938, 2021.

Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. In *2013 IEEE International Conference on Robotics and Automation*, pages 4054–4061. IEEE, 2013.

Rowan McAllister, Gregory Kahn, Jeff Clune, and Sergey Levine. Robustness to out-of-distribution inputs via task-aware generative uncertainty. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2083–2089. IEEE, 2019.

Q. Nguyen and K. Sreenath. Robust safety-critical control for dynamic robotics. *IEEE Transactions on Automatic Control*, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Zengyi Qin, Dawei Sun, and Chuchu Fan. Sablas: Learning safe control for black-box dynamical systems. *IEEE Robotics and Automation Letters*, 7(2):1928–1935, 2022.

Charles Richter and Nicholas Roy. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems*. Cambridge, MA, 2017.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3717–3724. IEEE, 2020.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio A Vela. Synthesis of control barrier functions using a supervised machine learning approach. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7139–7145. IEEE, 2020.

Andrew J. Taylor, Victor D. Dorobantu, Sarah Dean, Benjamin Recht, Yisong Yue, and Aaron D. Ames. Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty. In *IEEE Conference on Decision and Control*, pages 6469–6476, 2021.

Herke Van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3928–3934. IEEE, 2016.

Han Wang, Kostas Margellos, and Antonis Papachristodoulou. Safety verification and controller synthesis for systems with input constraints. *arXiv preprint arXiv:2204.09386*, 2022.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Albert Wilcox, Ashwin Balakrishna, Brijen Thananjeyan, Joseph E Gonzalez, and Ken Goldberg. Ls3: Latent space safe sets for long-horizon visuomotor control of sparse reward iterative tasks. In *Conference on Robot Learning*, pages 959–969. PMLR, 2022.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, pages 7444–7453. PMLR, 2019.