Out-of-Distribution-Aware Control Barrier Estimation

Hongzhan Yu
Computer Science & Engineering
UC San Diego
Email: hoy021@ucsd.edu

Sicun Gao
Computer Science & Engineering
UC San Diego
Email: sicung@ucsd.edu

***Abstract*—Learning-based synthesis of control barriers is an emerging approach to certifying safety for robotic systems. Yet, its effectiveness hinges on self-annotation, i.e., how to assign provisional safety labels to the states with no expert ground truth. The prevailing pipeline annotates each unlabeled sample by forward-simulating it for a short horizon and trusting the network predictions along that rollout, a procedure that is unreliable when the model has not yet generalized. This paper introduces an out-of-distribution-aware self-annotation framework that conditions every provisional label on both the predicted barrier value and a calibrated OOD score measuring how closely the query state lies on the network's training manifold. We conduct hardware experiments to evaluate the proposed method. With a limited amount of real-world data, it achieves state-of-the-art performance for static and dynamic obstacle avoidance, demonstrating statistically safer and less conservative maneuvers compared to existing methods.**

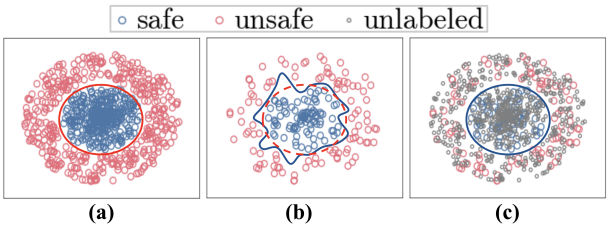

Fig. 1. Visualizations on toy datasets, to illustrate the motivation for utilizing unlabeled data. **(a)** With sufficient labeled data, the model can accurately capture the safety boundary. **(b)** When labeled data is limited, the learned boundary often misclassifies the safe and unsafe regions of the system. **(c)** Unlabeled data is generally more accessible than labeled data. Our approach leverages unlabeled data, along with the limited labeled data, to capture the CBF landscape that best adhere to the constraints inherent in the data.

I. Introduction

Control Barrier Functions (CBFs) offer a rigorous framework for certifying the forward-invariance of a prescribed safe set in nonlinear robotic systems [1, 2]. Closed-form construction of such a barrier is seldom tractable for high-dimensional dynamics. Recent work has turned to data-driven CBF synthesis, wherein a neural network is trained on empirical trajectories [17, 18]. Although the learning paradigm injects vulnerabilities to approximation error and distribution shift, it extends barrier certificates to previously intractable problems such as large-scale multi-robot coordination [23, 26, 27] and dense-crowd pedestrian avoidance [25].

A central obstacle in data-driven CBF synthesis is self-annotation, i.e., assigning provisional safety labels to samples that lack expert supervision. In online setting where learning and control proceed concurrently, states encountered just before a safety violation cannot be labeled with certainty because their true safety depends on future, unknown inputs. Offline pipelines also face a complementary dilemma: expert-labeled demonstrations are scarce, whereas unlabeled trajectories are easy to generate. The challenge in exploiting this abundant but uncertain data without compromising the certificate resorts to self-annotation. Mislabeling even a small subset of unlabeled samples can invalidate the certificate learning. Standard pipelines attempt to mitigate the issue by forward-simulating each unlabeled states for a short horizon and trusting the network's predictions [23, 27], but this strategy is fragile when the model has not yet generalized.

To address this fragility, we propose to incorporate out-of-distribution (OOD) awareness into the annotation loop. Each candidate label is accepted only when the predicted safety value is accompanied by a calibrated OOD score indicating that the state lies within the network's training manifold. States deemed OOD are withheld from training, preventing the propagation of erroneous labels. This paper is organized as follows. After reviewing related work and preliminaries (Sections II-III), we present the OOD-aware barrier learning algorithm (Section IV) and demonstrate its effectiveness in both simulation and hardware experiments (Section V). Section VI concludes the paper.

II. Related Work

Control Barrier Function. Control barrier functions (CBF) [3] aim to ensure control safety in dynamical systems by imposing value-landscapes to render the safe set forward invariant. The key point is to enforce the derivative of the CBF to satisfy Lyapunov conditions [13]. Traditional CBFs are manually designed based on domain-specific knowledge of the system, making them unsuitable for systems with complex dynamics or high uncertainty [1, 2].

Learning-based methods have been introduced to construct data-driven CBF candidates [20, 19, 15, 7, 17, 25, 26, 23] from data. Online algorithms learn CBFs by interacting with, or sampling from, the controlled system. In [20], the authors learn barrier certificates to derive the safe region of an unknown

control-affine system. They propose an adaptive sampling algorithm to iteratively refine the CBF candidate on the states that have high uncertainty. [15] studies the multi-agent control problem. They jointly learn the barrier certificates alongside the multi-agent control policy, while regulating the policy based on CBF. [7] develops a model-based approach to learn control Lyapunov barrier functions based on stability and safety specifications. The training state are sampled uniformly from the state space. Offline algorithms learn CBFs without new data during the learning. [18] proposes an incremental learning of a set of linear parametric CBFs from human demonstrations. In [17], the authors present an approach to synthesize local valid CBFs for control-affine systems with known but nonlinear dynamics. The expert demonstrations contain only safe trajectories collected with a fixed nominal controller.

Out-of-distribution Analysis. Out-of-distribution (OOD) analysis is an emerging topic of machine learning that examines the distribution shifts where test data diverges from the training data distribution [22]. Unsupervised representation learning methods focus on learning domain-agnostic features from unlabeled data [14, 28, 9, 5]. However, these methods can introduce bias, if the OOD domain distributions overlap with the unlabeled data distribution [24]. Supervised learning methods incorporate implicit domain labels from both in-distribution and OOD data [4, 21]. While these methods are often more accurate due to the additional information, they may not generalize well to OOD examples that differ significantly from those seen during training.

III. Preliminary

We consider discrete-time system dynamics $\dot{x}(t) = f(x(t), u(t))$ where $x(t)$ takes values in an n-dimensional state space $\mathcal{X} \subseteq \mathbb{R}^n$, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control vector, and $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is a Lipschitz-continuous vector field. We allow f to be generally nonlinear and not control-affine. Consider an unsafe region of the state space $\mathcal{X}_u \subset \mathcal{X}$ where safety constraints are violated. A subset of the state space $\mathsf{Inv} \subseteq \mathcal{X}$ is control invariant, if for any initial state $x(0) \in \mathsf{Inv}$ and any $t > 0$, we have $x(t) \in \mathsf{Inv}$. Namely, any trajectory that starts in the invariant set Inv stays in Inv forever. CBFs are scalar functions whose zero-superlevel set is a control invariant set within the safe region of the system, and whose spatial gradients can be used to enforce the invariance.

Definition 1 (Control Barrier Functions [3]): Let $B : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable function. The Lie derivative of B over f is defined as:

$$L_{f,u}B(x) = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} \cdot \frac{\partial x_i}{\partial t} = \langle \nabla_x B(x), f(x,u) \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. The Lie derivative measures the change of B along the system dynamics under control u. If the zero-superlevel set of B, i.e. $\mathcal{C} = \{x \in \mathcal{X} : B(x) \geq 0\}$, is disjoint from the unsafe region of the system, i.e. $\mathcal{C} \cap \mathcal{X}_u = \emptyset$. And if for any safe state $x \in \mathcal{C}$ and an extended class-$\mathcal{K}_\infty$ function $\alpha(\cdot)$ [11]:

$$\max_{u \in \mathcal{U}} L_{f,u}B(x) \geq -\alpha(B(x)). \tag{2}$$

Then B is a control barrier function (CBF), and its zero-superlevel set $\mathcal{C}$ is control invariant.

Out-of-distribution (OOD) detection examines whether a query point lies on the data manifold that a neural network was trained on *(in-distribution)* or depart from it *(out-of-distribution)*. Following Charoenphakdee et al. [5], we implement OOD detection as a binary classification-with-rejection problem governed by Chow's decision rule [6]. Let $\mathcal{X}$ be the input space and $\mathcal{Y} = \{-1, +1\}$ be the label set. For a fixed threshold $c \in (0,1) \subseteq \mathbb{R}$, we first train a probabilistic classifier $P : \mathcal{X} \to (0,1)$ so that the decision function $f_{c,P}(x)$ is:

$$f_{c,P}(x) = \begin{cases} +1, & P(x) > c, \\ -1, & P(x) \leq c. \end{cases} \tag{3}$$

To endow the system with a reject option, i.e., the ability to abstain when the confidence is low, we adopt the two-head architecture proposed in [5]. Specifically, we define two classifiers that share all weights except their final layer: $P_1 : \mathcal{X} \to (0,1)$ and $P_2 : \mathcal{X} \to (0,1)$, with respective thresholds c and $1-c$. A sample x is accepted as in-distribution only if both heads agree:

$$accept(x) = \Big[f_{c,P_1}(x) \cdot f_{1-c,P_2}(x) \Big] > 0. \tag{4}$$

Because the two heads are trained under complementary, cost-sensitive losses, they intentionally overfit to the training manifold in opposite directions. This makes the conjunction of their high-confidence regions sharply aligned with the true data support, so that points outside the support are rejected as OOD with high probability.

IV. Offline Learning of Barrier Critic

Consider that we have a well-defined CBF $B : \mathcal{X} \to \mathbb{R}$, and a discrete control system $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. For an arbitrary unlabeled state $o \in \mathcal{X}$ that does not violate safety, i.e. $o \notin \mathcal{X}_u$, if there exist controls at o that can lead the system to be in the zero-superlevel set of B:

$$\exists u \in \mathcal{U} \ s.t. \ B(f(o,u)) > 0, \tag{5}$$

then the state o satisfies the control invariant property, and thus, assigning a safe label to it must be correct. However, for the data-driven neural CBF models, following (5) can lead to incorrect annotations of the unlabeled. This is because if an unlabeled state o is uncovered by the training set, it is likely that neither is its one-step reachable set, i.e. $\mathcal{X}' = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U} \ s.t. \ f(o,u) = x\}$, covered fully. Thus, the model predictions on $\mathcal{X}'$ are not reliable in determining the safety of unlabeled samples.

In our work, we propose to label an unlabeled state o as safe if there exists a control $u \in \mathcal{U}$ such that:

$$B_\theta(\bar{x}) > 0 \ \textbf{\textit{and}} \ \bar{x} \text{ is } \textit{in-distribution} \text{ w.r.t. } \theta, \tag{6}$$

where $\bar{x} = f(o, u)$ and θ represents the parameters of neural CBF model. Intuitively, if we can derive such a control that leads the system to a *seen & safe* state, then there arises no concern about undermining the annotation steps due to the OOD samples. In the following sections, we describe the components for achieving the proposed idea.

A. Rejection-based Out-of-distribution Analysis

We employ [5] to determine whether a given input is in-distribution. Let the rejection model be denoted by $R_\Phi : X \to \mathbb{R}^2$, which outputs two-dimensional rejection scores for the given state input. Denote the rejection threshold by $c \in (0, 1) \subseteq \mathbb{R}$. Over the safe set $\mathbb{X}_s$ and the unsafe set $\mathbb{X}_u$, we optimize R_Φ to minimize the following objective:

$$\begin{aligned} L_\Phi(\mathbb{X}_s, \mathbb{X}_u) &= L_{\Phi_1,c}(\mathbb{X}_s, \mathbb{X}_u) + L_{\Phi_2,1-c}(\mathbb{X}_s, \mathbb{X}_u), \\ L_{\Phi_i,(\cdot)}(\mathbb{X}_s, \mathbb{X}_u) &= \frac{1}{|\mathbb{X}_s|} \sum_{x \in \mathbb{X}_s} [-R_{\Phi_i}(x) + (\cdot)]_+ \\ &\quad + \frac{1}{|\mathbb{X}_u|} \sum_{x \in \mathbb{X}_u} [R_{\Phi_i}(x) - (\cdot)]_+, \end{aligned} \tag{7}$$

where $R_{\Phi_i}(x)$ denotes the ith score from $R_\Phi(x)$ and $[\cdot]_+ = \max(\cdot, 0)$. When the rejection model is well trained, we say that the state $x \in \mathcal{X}$ is an in-distribution sample if:

$$R_{\Phi_1}(x) > c \textbf{ \textit{and} } R_{\Phi_2}(x) > 1 - c, \tag{8}$$

implying no disagreement between the two rejection scores.

B. Actor Model Learning

The rejection model enables us to classify if a given state is in-distribution or not. However, to realize the annotation steps proposed in (6), we must be able to efficiently determine what controls to attempt at one unlabeled state.

We achieve this by learning an actor model $\pi_\Theta : \mathcal{X} \to \mathcal{U}$ that captures the maximally-safe, in-distribution control for the given state. The term 'maximally-safe' is with respect to the CBF landscape, accounting for the maximal increase to the learned CBF score led by the control. Denote the CBF model by $B_\theta : \mathcal{X} \to \mathbb{R}$. With the rejection model R_Φ and the parameter c, we aim at solving the following optimization problem with the actor at an arbitrary state $x \in \mathcal{X}$:

$$\begin{aligned} &\arg\max_{u \in \mathcal{U}} \; B_\theta(f(x, u)), \\ &\textit{s.t. } R_{\Phi_1}(f(x, u)) > c \textit{ and } R_{\Phi_2}(f(x, u)) > 1 - c. \end{aligned} \tag{9}$$

Consider that we obtain the control u^* by solving (9) at an unlabeled state o. If following u^* at o cannot satisfy (6), then no less safe control can satisfy it either. Therefore, we can label o as unsafe without evaluating any other controls. Given a training batch $\mathbb{X}$, we optimize Θ by minimizing:

$$\begin{aligned} L_\Theta(\mathbb{X}) = \frac{1}{|\mathbb{X}|} \sum_{x \in \mathbb{X}} \Bigg[&- B_\theta\Big(f(x, \pi_\Theta(x))\Big) \\ &+ [-R_{\Phi_1}\Big(f(x, \pi_\Theta(x))\Big) + c]_+ \\ &+ [-R_{\Phi_2}\Big(f(x, \pi_\Theta(x))\Big) + 1 - c]_+ \Bigg]. \end{aligned} \tag{10}$$

Algorithm 1 Neural CBF with Barrier Critic (NCBF-BC)

Input: labeled sets D_s and D_u, unlabeled set D_{ul}, training iteration T, annotation start iteration T_a

Initialize the models and data buffer
for $t = 1$ ... T **do**
  Sample labeled batches $\mathbb{X}_s \subseteq D_s$ and $\mathbb{X}_u \subseteq D_u$
  if $(t \geq T_a)$ **then**
    Sample an unlabeled batch $\mathbb{X}_{ul} \subseteq D_{ul}$
    $\mathbb{X}_{ul,s}$, $\mathbb{X}_{ul,u} \leftarrow$ *Annotate*($\mathbb{X}_{ul}$)
    $\mathbb{X}_s \leftarrow \mathbb{X}_s \cup \mathbb{X}_{ul,s}$, $\mathbb{X}_u \leftarrow \mathbb{X}_u \cup \mathbb{X}_{ul,u}$
  end if
  Model *updates* over $\mathbb{X}_s$ and $\mathbb{X}_u$: R_Φ with (7), π_Θ with (10), B_θ with (11)
end for
return B_θ, R_Φ, π_Θ

Function *Annotate*($\mathbb{X}$):
  $\mathbb{X}_s \leftarrow \{\}$, $\mathbb{X}_u \leftarrow \{\}$
  for x in $\mathbb{X}$ **do**
    $\bar{x} \leftarrow f(x, \pi_\Theta(x))$
    if $B_\theta(\bar{x}) > 0$ **and** $R_\Phi(\bar{x})$ satisfies (8) **then**
      $\mathbb{X}_s \leftarrow \mathbb{X}_s \cup \{x\}$
    else
      $\mathbb{X}_u \leftarrow \mathbb{X}_u \cup \{x\}$
  end for
  return $\mathbb{X}_s$, $\mathbb{X}_u$
EndFunction

Unlike Reinforcement Learning (RL) methods [12, 8], we do not rely on the actor to generate controls at execution time. Instead, the actor is used soly as an auxiliary model to shape the barrier landscape during training.

C. Overall Pipeline

We now discuss the learning pipeline of the CBF model. We write the CBF model as $B_\theta : \mathcal{X} \to \mathbb{R}$. Given a safe batch $\mathbb{X}_s$ and an unsafe batch $\mathbb{X}_u$, we optimize θ by minimizing:

$$\begin{aligned} &L_\theta(\mathbb{X}_s, \mathbb{X}_u) \\ &= \left(\frac{1}{|\mathbb{X}_s|} \sum_{x \in \mathbb{X}_s} [-B_\theta(x)]_+ \right) + \left(\frac{1}{|\mathbb{X}_u|} \sum_{x \in \mathbb{X}_u} [B_\theta(x)]_+ \right) \\ &+ \frac{1}{|\mathbb{X}_s|} \sum_{x \in \mathbb{X}_s} \Big[- \Big\langle \nabla_x B_\theta(x), \nabla_x f(x, \pi_\Theta(x)) \Big\rangle - \alpha \Big(B_\theta(x) \Big) \Big]_+ . \end{aligned} \tag{11}$$

The first two terms enforce $B_\theta(x)$ to take positive values on safe states and negative values on unsafe states, respectively. The third term optimizes the model to satisfy the Lie derivative condition of CBF in (2).

Unlike prior work [17, 25, 23], which optimizes the Lie derivative condition over the safe controls from data, we optimize it over the controls generated by the actor π_Θ. In fact, optimizing with maximally-safe controls more closely follows the original CBF definition (2) which applies a $\max$ operator over the control space on the Lie derivative. In Section V-A, we show that incorporating the actor allows

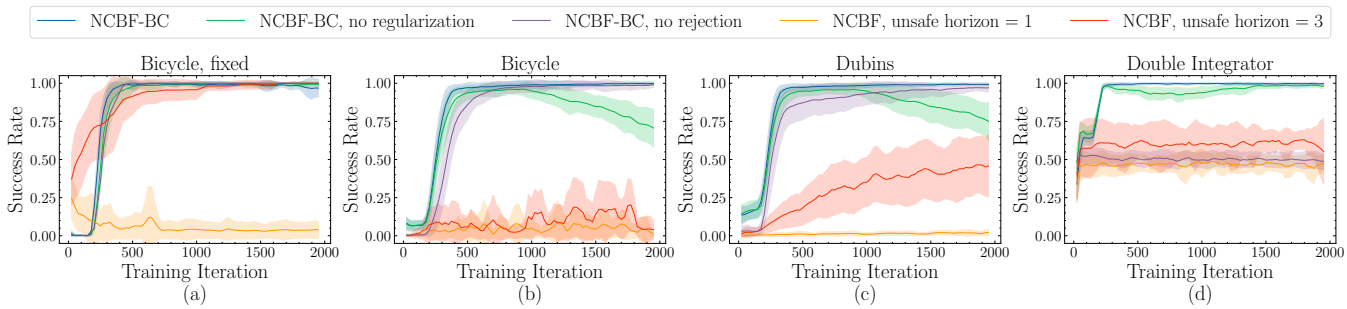

Fig. 2. Simulation experiments for static obstacle avoidance with different dynamics models of ego-robot. Evaluation metric is the mean success rate where we follow Algorithm 2 to derive the controls based on trained models, and perform evaluations over 100 randomized scenarios. When collecting the safe trajectories, we employ the potential-field controller(s) with **(a)** *fixed* and **(b)-(d)** *randomized* parameters.

for training data collected with diverse controllers without imposing any performance assumption on them.

We present the full procedures in Algorithm 1. Early in training, unlabeled data remain unannotated until a sufficient number of iterations have been completed (Line 4). This is to prevent false model estimations at the outset. To annotate an unlabeled state x, we unroll the dynamics function using the actor's control output to obtain the next state $\bar{x}$ (Line 14). We then label x as safe if and only if $\bar{x}$ is deemed safe with respect to the CBF model B_θ **and** in-distribution with respect to the rejection model R_Φ (Line 15).

D. Optimization Regularization

As the learning objective of the CBF model (11) enforces inequality constraints on the estimated landscape, the training process can suffer from the collapse problem similar to that reported in self-supervised learning [10]: as training proceeds, the magnitude of the learned landscape may shrink towards near-zero values while still violating the inequality constraints.

To alleviate the collapse issue, we optimize (11) using the surrogate CBF values $\bar{B}_\theta$ defined as follows:

$$\bar{B}_\theta(x) = B_\theta(x)/\mathbb{E}_{x\in\bar{\mathbb{X}}}\big[B_\theta(x)\big], \tag{12}$$

where $\bar{\mathbb{X}} \subseteq \mathbb{X}_s$ is a subset of the safe set sampled in advance. We do not detach the gradient of the denominator in (12) with respect to model parameters θ, which helps to elevate the overall magnitude of barrier landscape whenever it begins to collapse toward zero.

V. Experiments

We evaluate the proposed algorithm in both simulation and real-world experiments. To derive safety-critical controls from the learned models, we follow Algorithm 2 which requires a heuristic goal-driven metric to rank controls according to task completion progress.

A. Simulation Experiments

We focus the simulation evaluations on obstacle avoidance. We generate training trajectories using potential-field controllers with either *fixed* or *randomized* parameters. A trajectory is considered safe if no collision occurs. If a trajectory ends in a collision, we add the collision state to the unsafe set, and its preceding segment (of unlabel horizon τ) to the unlabeled set. In our simulation environments, where the time-step is discretized at $\triangle t = 0.2$ second, we set the unlabel horizon $\tau = 9$.

We consider three different ego-robot dynamics including Double Integrator, the Dubins, and the Bicycle models. For each dynamics model, the system consists of vehicle linear and angular velocities and yaw angles besides the coordinates. All the neural network models are 2-layer Tanh networks with 128 hidden neurons per layer. We use rejection parameter $c = 0.1$, and perform optimization regularization with subset size 1000. We employ the linear mapping $\alpha(x) = \kappa \cdot x$ with $\kappa = 0.1$. The algorithm runs for 2000 iterations in total, while the annotation of unlabeled data starts at the 200th iteration. We use the orientation towards the goal as the heuristic goal-driven metric.

Comparisons with Baseline Methods. Figure 2 demonstrates the experiment results. The baseline is the standard method for learning CBFs as in [7, 17, 16, 25, 23], denoted by *Neural CBF* (NCBF). The *Unsafe horizon* defines the number of states near the end of failure trajectories which we label as unsafe, but only when training the baseline.

First, in Figure 2(b)-(d), we show that the baseline underperforms when training trajectories are generated by multiple controller polices. This occurs because it is incorrect to optimize the Lie derivative condition (2) of CBF over all the provided safe controls. Our method can handle training sets collected by a diverse range of sub-optimal controllers, as it optimizes the Lie derivative condition only along the maximally-safe control input, allowing those less conservative inputs to violate the inequality constraint in (2). Furthermore, Figure 2(a) shows that using training data that is generated by a fixed expert policy, our OOD-aware method still delivers the fastest learning convergence.

Across all experiments, the proposed method incorporating both regularization and rejection-based annotation performs the best in terms of learning rates and training stability. For the Bicycle and Dubins models, the regularization technique effectively prevents collapse, thereby avoiding divergence during training. The Double Integrator does not exhibit collapse issues, regardless of whether regularization is applied. Because it has simpler dynamics, the optimization for satisfying the CBF

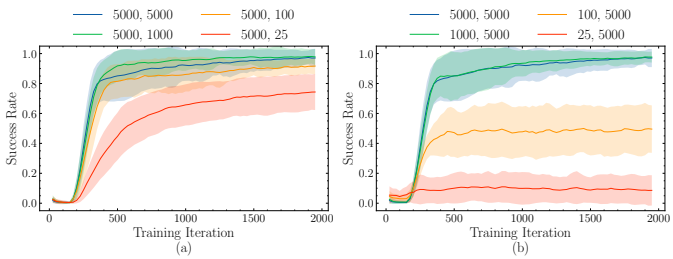

Fig. 3. Ablation experiments on the ratio between training labeled and unlabeled set sizes for Bicycle. The two numbers in each label are the sizes of labeled and unlabeled demonstration sets, respectively. For instance, the blue curve in (a) refers to the setting with 5000 labeled safe & unsafe states, and 5000 unlabeled states.

conditions converges before collapse can occur. Meanwhile, disabling the rejection-based annotation noticeably slows convergence. In particular, for the Double Integrator, the CBF quickly overfits to the available labeled data, weakening the effectiveness of the annotation process that relies only on the learned CBF scores.

Ratio of Labeled Data. We conduct ablation experiments on the Bicycle model to investigate how the ratio of training set size impacts the proposed method. In Fig. 3(a), we vary the size of unlabeled set while fixing the labeled size, showing that the learning can be quickly stimulated even with small amounts of unlabeled data. This is because the labeled states that are certainly safe may be derived from conservative policies whose safety rules deviate from the optimal safety boundary. Meanwhile, the unlabeled trajectories with uncertain safety often involve aggressive controls that may exceed the optimal safety boundary, and thus carry more useful information. In Fig. 3(b), we fix the unlabeled size while varying the size of labeled set. With sufficient unlabeled data provided, there appears to be a threshold to the labeled size beyond which the learning rates become indifferent.

Algorithm 2 Control using NCBF-BC

Input: state x, CBF model B_θ, rejection model R_Φ, goal-driven metric $\mathcal{G} : X \to \mathbb{R}$, sample size N

Sample control candidates $\boldsymbol{a} = [a_1, a_2, ..., a_N]$
$\mathbf{g} \leftarrow []$
for a_i in $\boldsymbol{a}$ **do**
  Unroll the dynamics $\bar{x} = f(x, a_i)$
  if $B_\theta(\bar{x} < 0)$ **or** $R_\Phi(\bar{x})$ does not satisfy (8) **then**
    Remove a_i from $\boldsymbol{a}$
  else
    Evaluate the goal-driven score, $\mathbf{g} = \mathbf{g} \cup \{\mathcal{G}(\bar{x})\}$
  end if
end for
if $\boldsymbol{a}$ is now empty **then**
  return *Error - no safe control found*
end if
return control candidate from $\boldsymbol{a}$ with the maximal score

B. Hardware Experiments

In this section, we discuss hardware experiments on dynamic obstacle avoidance (Figure 5). The experiments on static obstacles (Figure 4 Right) are showcased in supplementary video but are not discussed in the paper. The platform utilized in our experiments is the Freight (Figure 4 Left), a research variant from Fetch Robotics. We cap the velocity of the robot at 0.22 m/s. To train our model, we collected 40-minutes of demonstrations by manually driving the robot around pedestrians, deliberately splitting the data into roughly 15-minutes of successful and 25-minutes of failure trajectories. For applying the proposed method, we discretize the time-step to be $\Delta t = 0.15$ second, and employ unlabel horizon $\tau = 9$.

The system state space for dynamic obstacle avoidance is 11-dimensional, consisting of robot coordinates, yaw and velocity information, and 3-step past state history of individual pedestrian. All the neural models are 2-layer Tanh networks with 256 hidden neurons per layer. We perform the learning for 5000 iterations, initiating the annotation steps over unlabeled data at the 500th iteration. All other training parameters match those used in simulation experiments. When optimizing (11) to enforce the Lie derivative condition of CBF, we only leverage the derivative of ego-robot dynamics, while taking from data the pedestrian movements at the future timestamps.

Table I presents the quantitative results. Besides *NCBF*, we include one potential-field controller using a repulsive range of 1.5 meter. We further compare against both the ROS1 MoveBase (MPC-based) navigation stack and the ROS2 Nav2 stack. Offline Deep RL algorithms cannot be directly applied, as reward labels that can accurately reproduce the given controls are typically unavailable in real-world data.

First, the proposed method achieves the highest success rate. The failures with our method are always due to unfamiliar pedestrian movements that deviate from the training data. Second, we show that the proposed method completes the scenarios with the highest mean velocity, while maintaining the lowest distance to the pedestrians without violating safety. This showcases the robustness of the learned safety boundary which allows us to select the controls that are performative, or even aggressive, yet safe. Third, the performance of potential-

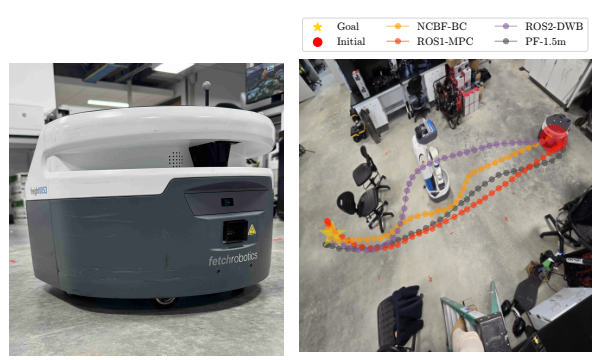

Fig. 4. **Left**: Freight robot. **Right**: Trajectories generated by different controllers.

	Success Rate (%)	Mean Path Length (meter)	Mean Completion Time (sec)	Mean Velocity (meter/sec)	Minimal Distance to Obstacles (meter)
NCBF-BC (ours)	**93.3**	7.23	36.46	**0.21**	**0.578**
NCBF	46.7	7.60	35.49	**0.21**	0.610
PF-1.5m	80.0	7.30	**35.06**	0.20	0.668
ROS1-MPC	80.0	7.75	44.65	0.17	0.697
ROS2-DWB	86.7	**6.68**	40.26	0.17	0.677

TABLE I
REAL-WORLD EXPERIMENTS FOR DYNAMIC OBSTACLE AVOIDANCE OVER 30 RUNS.

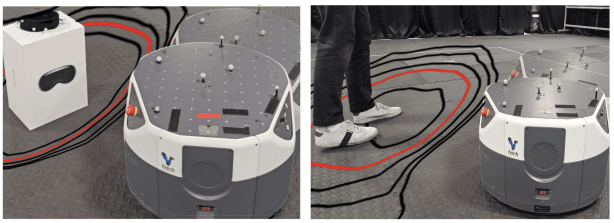

Fig. 5. Visualization of the learned CBF landscapes. Note that the CBF model trained for dynamic obstacle avoidance exhibits a wider gap between level sets, which reflects the need to initiate collision avoidance further from dynamic obstacles compared to static ones.

field controller degrades when there involve more pedestrians surrounding the robot. Oscillation behaviors are observed with potential-field controller in our experiments. Last, the NCBF baseline shows sub-optimal performance, producing strange looping behaviors and frequently taking unnecessarily long paths when pedestrians are present. Since data were collected via manual control, states could be reached with controls of varying levels of conservativeness. Consequently, the baseline's training objective forces the CBF to accommodate the most conservative control among the provided examples. Moreover, the NCBF baseline under-utilize failure trajectories, especially the uncertain states preceding the collisions, thereby limiting the amount of data it can effectively leverage.

VI. CONCLUSION

This paper integrates out-of-distribution (OOD) detection into the self-annotation step of data-driven CBF synthesis. Experiments show that the OOD-aware pipeline produces safer and less conservative barriers than existing offline methods by exploiting unlabeled trajectories more effectively. Future work will adapt the scheme to online learning and explore stronger OOD metrics to further boost reliability.

REFERENCES

[1] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278. IEEE, 2014.

[2] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[5] Nontawat Charoenphakdee, Zhenghang Cui, Yivan Zhang, and Masashi Sugiyama. Classification with rejection based on cost-sensitive classification. In *International Conference on Machine Learning*, pages 1507–1517. PMLR, 2021.

[6] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16 (1):41–46, 1970.

[7] 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.

[8] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[9] Sivan Harary, Eli Schwartz, Assaf Arbelle, Peter Staar, Shady Abu-Hussein, Elad Amrani, Roei Herzig, Amit Alfassy, Raja Giryes, Hilde Kuehne, et al. Unsupervised domain generalization by learning a bridge across domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5280–5290, 2022.

[10] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[11] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002. URL https://cds.cern.ch/record/1173048. The book can be consulted by contacting: PH-AID: Wallet, Lionel.

[12] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforce-

ment learning. *arXiv preprint arXiv:1509.02971*, 2015.

[13] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

[14] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International conference on machine learning*, pages 7313–7324. PMLR, 2021.

[15] Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436*, 2021.

[16] Zhizhen Qin, Tsui-Wei Weng, and Sicun Gao. Quantifying safety of learning-based self-driving control using almost-barrier functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12903–12910. IEEE, 2022.

[17] 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.

[18] Matteo Saveriano and Dongheui Lee. Learning barrier functions for constrained motion planning with dynamical systems. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 112–119. IEEE, 2019.

[19] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control*, pages 708–717. PMLR, 2020.

[20] Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2460–2465. IEEE, 2018.

[21] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*, 2022.

[22] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pages 1–28, 2024.

[23] Chenning Yu, Hongzhan Yu, and Sicun Gao. Learning control admissibility models with graph neural networks for multi-agent navigation. In *Conference on Robot Learning*, pages 934–945. PMLR, 2023.

[24] Han Yu, Xingxuan Zhang, Renzhe Xu, Jiashuo Liu, Yue He, and Peng Cui. Rethinking the evaluation protocol of domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21897–21908, 2024.

[25] Hongzhan Yu, Chiaki Hirayama, Chenning Yu, Sylvia Herbert, and Sicun Gao. Sequential neural barriers for scalable dynamic obstacle avoidance. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11241–11248. IEEE, 2023.

[26] Songyuan Zhang, Kunal Garg, and Chuchu Fan. Neural graph control barrier functions guided distributed collision-avoidance multi-agent control. In *Conference on Robot Learning*, pages 2373–2392. PMLR, 2023.

[27] Songyuan Zhang, Oswin So, Kunal Garg, and Chuchu Fan. Gcbf+: A neural graph control barrier function framework for distributed safe multi-agent control. *IEEE Transactions on Robotics*, 2025.

[28] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. Towards unsupervised domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4910–4920, 2022.