2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

ConBaT: Control Barrier Transformer for Safe Robot Learning from Demonstrations

Yue Meng[1], Sai Vemprela[2], Rogerio Bonatti[2], Chuchu Fan[1] and Ashish Kapoor[2]

Abstract— **Large-scale self-supervised models have recently revolutionized our ability to perform a variety of tasks within the vision and language domains. However, using such models for autonomous systems is challenging because of safety requirements: besides executing correct actions, an autonomous agent must also avoid the high cost and potentially fatal critical mistakes. Traditionally, self-supervised training mainly focuses on imitating previously observed behaviors, and the training demonstrations carry no notion of which behaviors should be explicitly avoided. In this work, we propose Control Barrier Transformer (ConBaT), an approach that learns safe behaviors from demonstrations in a self-supervised fashion. ConBaT is inspired by the concept of control barrier functions in control theory and uses a causal transformer that learns to predict safe robot actions autoregressively using a critic that requires minimal safety data labeling. During deployment, we employ a lightweight online optimization to find actions that ensure future states lie within the learned safe set. We apply our approach to different simulated control tasks and show that our method results in safer control policies compared to other classical and learning-based methods such as imitation learning, reinforcement learning, and model predictive control.**

I. INTRODUCTION

Mobile robots are finding increasing use in complex environments through tasks such as autonomous navigation, delivery, and inspection [41], [26]. Any unsafe behavior, such as collisions in the real world, can potentially result in catastrophic outcomes. Hence, robots are expected to act in a safe, reliable manner while achieving the desired goals. Yet, learning safe behaviors comes with several challenges. Primarily, notions of safety are often indirectly found in datasets, as it is customary to show optimal actions (what the robot should do) as opposed to demonstrating failures (what to avoid). In fact, defining explicit safety criteria in real-world scenarios is complex and requires domain knowledge [28], [14], [32]. In addition, the binary safe/unsafe outcome inferred from observations might be insufficient to make safe decisions - when it realizes it is unsafe, it might be too late to recover.

Classical safe planning methods often treat safety as constraints and achieve the task goal by solving the optimization problems, which rely on carefully crafted models and expensive parameters tuning [64], [58], [56]. While those methods are easy to set up for simple dynamics and safety concepts, challenges in translating safety definitions into rules hinder the deployment of classical methods in complex settings and also make such planners prone to adversarial attacks [60].

There also exist safe learning-based approaches such as reinforcement learning, imitation learning [16], [57], and model-based methods via reachability analysis and control barrier functions [30], [36]. While expert demonstrations may reveal one way to solve a particular task, they do not often reflect which unsafe behaviors should be avoided. We can draw similarities and differences with other domains: natural language (NL) and vision models can learn to generate grammatically correct text or temporally consistent future images by following patterns consistent with the training corpus. However, for robotics, notions of safety are less evident from demonstrations. While the cost of a mistake is not fatal in NL and vision, disobeying the safety rules can have significant negative consequences for cyber-physical systems. Our paper aims to answer a fundamental question: how to learn a policy from demonstrations that is both effective for tasks and respects safety constraints?

Recently, the success of large language models [59], [15] has inspired the development of a class of Transformer-based models for decision-making which uses auto-regressive losses over sequences of demonstrated state and action pairs [48], [12]. While such models are able to learn task-specific policies from expert data, they lack a clear notion of safety and are unable to avoid unsafe actions explicitly. Our work builds upon this paradigm of Transformers applied to perception-action sequences and proposes a method to learn policies in a safety-critical fashion.

Our method, named Control Barrier Transformer (ConBaT), takes inspiration from the control barrier functions (CBF) from control theory [2]. Our architecture consists of a causal Transformer augmented with a safety critic that mimics a CBF to evaluate safety. Instead of complex hand-crafted safety rules, we only require binary labels to tell whether demonstrations are safe. This control barrier critic then learns to map observations to a continuous safety score, inferring safety in a self-supervised way. A lightweight optimization scheme operates on the critic values to minimally modify the proposed action and result in a safer policy, which is inspired by optimal control and enabled by the fully differentiable fabric of the model. Unlike classical CBF, our critic operates in embedding space instead of state space, making it applicable to a wide variety of systems.

Our contributions are: (1) We propose ConBaT, a causal Transformer architecture with a differentiable safety critic

*This work was done when Yue was an internship at Microsoft Research.

[1]Yue Meng and Chuchu Fan are with Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 70 Vassar St, Cambridge, MA 02139, USA {mengyue, chuchu}@mit.edu

[2]Sai Vemprela, Rogerio Bonatti and Ashish Kapoor were with the Autonomous Systems and Robotics Group at Microsoft Research, 14820 NE 36th St, Redmond, WA 98052, USA {saihv, rbonatti, akapoor}@microsoft.com

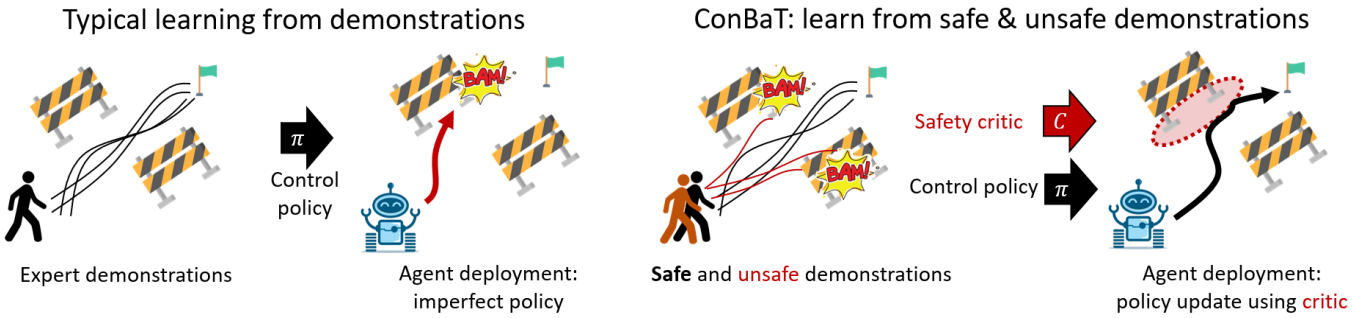


Fig. 1: (Left) An agent trained to imitate expert demonstrations may focus on the result of the task without explicit notions of safety. (Right) Our method ConBaT learns a safety critic on top of the control policy and uses this control barrier critic to optimize the policy for safe actions actively.

which can be applied to safety-critical applications. (2) We conduct experiments in a simplified F1 simulator and a 2D LiDAR navigation and ConBaT results in lower collision rates and longer safe trajectories compared to baselines. (3) ConBaT can learn novel safety concepts (e.g., avoid driving in straight lines) quickly with minimal data labeling.

II. Related work

Learning safe policies from data / rewards. Imitation Learning (IL) is a popular method for policy learning, and requires expert demonstrations [50], [5], [44], [7], [11], [24]. The most naive IL approach is behavioral cloning (BC) [6], which directly learns the mapping from states to actions using supervised learning from expert data. Such simple applications of IL cannot generalize to the out-of-distribution scenarios induced by on-policy deployment, and often requires additional training routines [49]. Inverse reinforcement learning (IRL) [40], [39] can mitigate this challenge, as it seeks to recover the original expert's cost function. Alternatively, one may use generative adversarial imitation learning (GAIL) [31], [10], which employs a discriminator to differentiate an expert's policy from other behaviors generated by a policy network. When it comes to safety, however, none of the above methods incorporate the concept of safety directly into their learning procedures.

Reinforcement learning can enable safe policy learning through rewards that not only encourage good behaviors but also penalize unsafe states. [9] use an augmented reward for traffic simulation to guide safe imitation, and we find numerous examples of safe policy design [35], [8], [21], [34]. However, manual reward shaping is laborious and nontrivial which often leads to undesired consequences. In comparison, our method with learned control barrier functions uses minimal safety labels.

Safe policy learning using control barrier functions. Control barrier function (CBF) is a classical concept to ensure system safety [45], [62], [3], [2]. Recently, several works have demonstrated learning safe policies based on existing CBFs [20], [13], constructing CBF jointly with the policy learning via Sum-of-Squares [61], leveraging SVMs [55] or Neural Networks [25], [47], [37], [22], [23]. However, these methods often assume access to raw state inputs and ground truth safety labels from the environment. Instead, ConBaT works on the embedding space akin to models that uses imagination [42], [63], learns from offline data, and employs an online optimization to achieve safety.

Safe policy learning with world models. ConBaT is aligned with the Mode-2 proposal in [33], where the agent makes action proposals, evaluates the costs from future predictions, and then plans for the next action. Predicting the next state based on the current state and action is fundamental to model predictive control [17]. Predicting future costs via the world model can be traced back to [51]. When it comes to high-dimension state/observation space, latent embedding dynamics are learned to achieve high performance in reinforcement learning [42], [63], and sequence predictions [27], [19], [38]. In our case, we learn a predictive critic on the embedding space with CBF conditions as guidance.

III. Methods

We consider the problem of learning safe control from demonstrations. An agent interacts with an environment to receive observations and then decides continuous-valued actions. For simplicity, we treat the observations the same as states while noting that actual states might not be evident from the data. The states could be low-dimension physical properties (e.g., positions, velocities) or high-dimensional sensor readings (e.g., LiDAR). Define a trajectory τ as a set of state-action pairs $\{(s_t, a_t)\}_{t=0}^{T}$. Our demonstrations are from two sets of trajectories: Σ_{s}, which are always safe, and Σ_{u}, which lead to unsafe terminal states. We aim to learn a safe policy $\pi_{\text{safe}} : s \to a$ that mimics the action in Σ_{s} while avoiding actions that lead to the unsafe states as in Σ_{u}.

A. Base architecture: Perception-Action Causal Transformer

The base architecture in ConBaT is the Perception-Action Causal Transformer (PACT) [12], which learns a pretrained representation that can be finetuned towards diverse robot tasks. In pretraining stage, it uses state-action pairs from expert demonstrations to autoregressively train a world model and a policy model. The main components are:

Tokenizer. The state and action tokenizers operate on raw observation and action data and learn to encode them as compact tokens: $T_s(s_t) \to s'_t$, $T_a(a_t) \to a'_t$, where $s', a' \in$

$\mathbb{R}^d$ with d being the token dimension. T_s and T_a can be neural networks (fully-connected networks, PointNet or Convolutional Neural Networks) or learnable matrices depending on the observation modalities.

Causal Transformer. A set of Transformer blocks operate upon a sequence of state and action tokens $s'_0, a'_0, ..., s'_T, a'_T$ and output a sequence of state and action embeddings $s^+_0, a^+_0, ..., s^+_T, a^+_T$ in the same dimension. A causal attention mask is applied to ensure that the Transformer generates the output embeddings at each timestep t only using the history state and action tokens from $[0, t]$.

Policy model. An action prediction head acts as a policy, operates on the output state embedding s^+_t at a given timestep t, and predicts the appropriate action: $\pi(s^+_t) \rightarrow \hat{a}_t$.

World model. This module operates on the state and action output embeddings from the current timestep, and predicts the next state embedding: $\phi(s^+_t, a^+_t) \rightarrow \hat{s}'_{t+1}$. This module serves as a regularizing mechanism during training.

B. Control barrier critic

On top of PACT, we design learnable modules that allow ConBaT to distinguish between safe and unsafe behaviors and to generate safe actions. The ConBaT architecture is shown in Fig. 2. We augment the transformer backbone with two trainable critics which predict safety scores for the current and future expected states. The first critic $C : s^+_t \rightarrow \hat{c}_t \in \mathbb{R}$ maps the current state embedding s^+_t to a real-valued safety score $\hat{c}_t$. The second critic $C_f : (s^+_t, a^+_t) \rightarrow \hat{c}_{t+1} \in \mathbb{R}$ estimates the future state safety score $\hat{c}_{t+1}$ based on the current state and action embeddings s^+_t, a^+_t. Here C_f can be interpreted as conjoining both a world model and safety critic within a single network.

To train these critics, we draw inspiration from the control barrier functions (CBF) in classical controls literature [2]. Given a system $\dot{s} = f(s, a)$ with f locally Lipschitz continuous and a safe set of states as $\mathcal{S}_s \subset \mathcal{S}$, if there exists a function $h : \mathcal{S} \rightarrow \mathbb{R}$ and a policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ satisfying:

$$h(s) \geq 0, \forall s \in \mathcal{S}_s; \quad h(s) < 0, \forall s \in \mathcal{S}_u = \mathcal{S}/\mathcal{S}_s;$$
$$\dot{h}(s) = \frac{\partial h(s)}{\partial s} \cdot f(s, \pi^*(s)) \geq -\alpha h(s) \tag{1}$$

with $\alpha > 0$, then h is called a control barrier function, and the policy π^* ensures any initial state starting from $\mathcal{S}_s$ will stay in $\mathcal{S}_s$ (forward invariant). The proof can be found in [3](Theorem 1). Thus, the policy π^* will guarantee all the states initialized from the safe set $\mathcal{S}_s$ to be always safe.

In practice, finding the perfect safe policy π^* is challenging due to input constraints or imperfect policy learning [47], [23]. One alternative is to learn the CBF on top of an imperfect policy π and then use an optimization routine (quadratic program [4], second-order cone program [18], gradient descent) to steer π to satisfy CBF constraints, thus achieving safety. ConBaT follows a similar philosophy where the critics are learned to approximate the CBF for a given (imperfect) policy π (from PACT) so that the CBF conditions can be satisfied in most cases, and then use back-propagation to rectify π to satisfy the CBF conditions in test. The supervision signals for learning the critic are just binary labels indicating states' safety, bypassing the need for hand-crafting complex signals. We call C and C_f Control Barrier Critics (CBC) and detail the learning process next.

C. Learning control barrier critic from demonstrations

As mentioned earlier, Σ_s and Σ_u represent the safe and unsafe demonstrations. By examining the states that constitute these demonstrations, we construct a collected safe set $\tilde{\mathcal{S}}_s \in \mathcal{S}_s$ and a collected unsafe set $\tilde{\mathcal{S}}_u \in \mathcal{S}_u$. As shown in Fig. 3, to form $\tilde{\mathcal{S}}_s$, we include all the states from the trajectories in Σ_s and the states from the first $(L - 2T)$ time steps from the trajectories in Σ_u where L is the expert trajectory length, and T is the time horizon that ConBaT takes as input. For $\tilde{\mathcal{S}}_u$, we only consider the terminal states from the trajectories in Σ_u (because those are the only 'unsafe' states we are certain of). The embeddings corresponding to $\tilde{\mathcal{S}}_s$ and $\tilde{\mathcal{S}}_u$ are $\tilde{\mathcal{S}}^+_s$ and $\tilde{\mathcal{S}}^+_u$ respectively.

During CBC learning, instead of considering all possible state embeddings, we only enforce the CBF conditions to be satisfied on $\tilde{\mathcal{S}}^+_s$ and $\tilde{\mathcal{S}}^+_u$. To satisfy Equation (1), we expect the critic values to be positive on the collected safe state embeddings $\tilde{\mathcal{S}}^+_s$, negative on the collected unsafe state embeddings $\tilde{\mathcal{S}}^+_u$, and to not decrease too fast for all collected state embeddings $\tilde{\mathcal{S}}^+ = \tilde{\mathcal{S}}^+_s \cup \tilde{\mathcal{S}}^+_u$. Note that $\tilde{\mathcal{S}}^+_s$ and $\tilde{\mathcal{S}}^+_u$ will be highly imbalanced because not only we have more safe demonstrations than the unsafe ones but also we only pick one state from each unsafe trajectory. However, as shown in future sections, our critics can be trained effectively even with a limited number of unsafe states.

We use three loss terms for training the CBC. First, a classification loss learns the safe set boundary:

$$\mathcal{L}_c = \mathop{\mathbb{E}}_{s^+_t \sim \tilde{\mathcal{S}}^+_s} \left[\sigma_+ \left(\gamma - C(s^+_t)\right)\right] + \mathop{\mathbb{E}}_{s^+_t \sim \tilde{\mathcal{S}}^+_u} \left[\sigma_+ \left(\gamma + C(s^+_t)\right)\right] \tag{2}$$

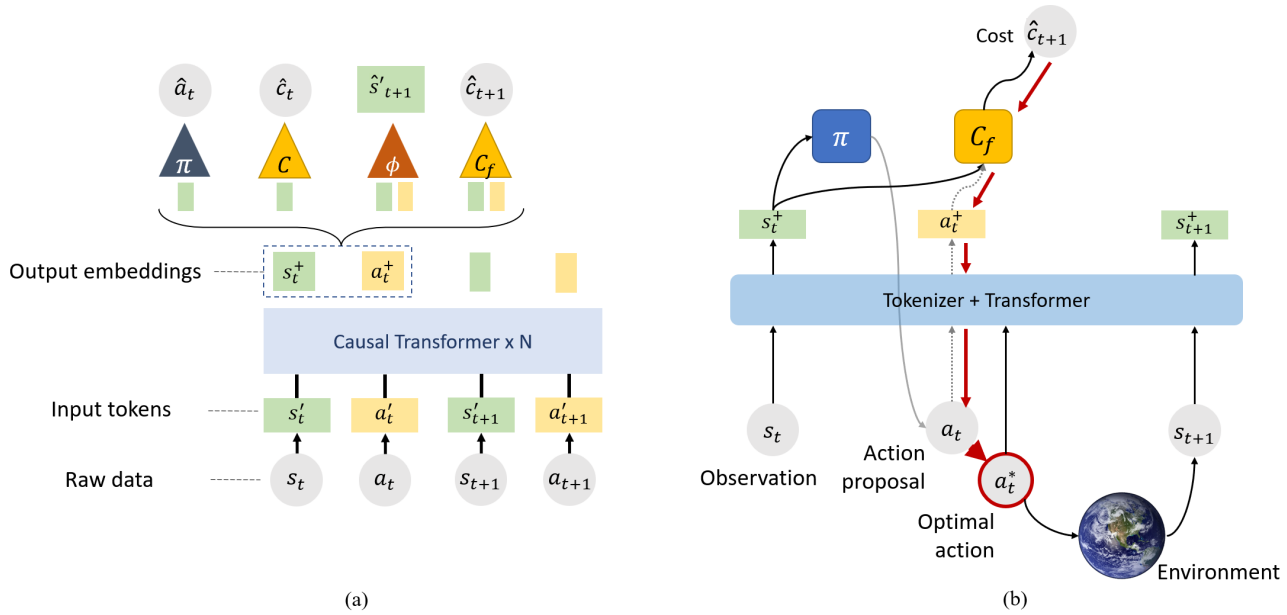
where the expectation $\mathbb{E}$ is approximated by computing the loss term on samples from the safe set $\tilde{S}^+_s$ and the unsafe set $\tilde{S}^+_u$ correspondingly. $\sigma_+(x) = \max(x, 0)$ and γ is a margin factor that ensures numerical stability in training. The second loss enforces smoothness on the CBC values over time:

$$\mathcal{L}_s = \mathop{\mathbb{E}}_{s^+_t \sim \tilde{\mathcal{S}}^+} \left[\sigma_+ \left((1 - \alpha)C(s^+_t) - C(s^+_{t+1})\right)\right] \tag{3}$$

where α controls the local decay rate. Note that this loss is asymmetrical as it only penalizes fast score decays but permits instantaneous increases, as a fast-improving safety level does not pose a problem. The final loss ensures consistency between the predictions of both critics C and C_f:

$$\mathcal{L}_f = \mathop{\mathbb{E}}_{s^+_t \sim \tilde{\mathcal{S}}^+} \left[\left(C_f(s^+_t, a^+_t) - C(s^+_{t+1})\right)^2\right] \tag{4}$$

The final loss is $\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_f \mathcal{L}_f$. We keep λ_c, λ_f to 1 and choose $\lambda_s = 5$ based on ablation studies (shown in the supplementary video). Though one could use the world model ϕ to get $\phi(s^+_t, a^+_t) \rightarrow \hat{s}'_{t+1}$ and uses the critic C to get future CBC score $C(\hat{s}'_{t+1})$, we found it helpful to use a separate critic head C_f to predict future CBC scores directly, as it improves the optimization process described next.



(a) (b)

Fig. 2: (a) The ConBaT architecture - a causal Transformer operates on state and action tokens (s', a') to produce embeddings (s^+, a^+). A policy head π computes actions given state embeddings, and a critic C computes a safety score. Both state and action embeddings are fed into a world model ϕ to compute the future state token, and by the future critic C_f to produce a future safety score. (b) The deployment process for ConBaT involves a feedback loop. The future critic evaluates action proposals from the policy head to check the safety. The red arrows show the flow of gradients that optimizes the safe action in a desired characteristic. The action a^* is used as the final command.

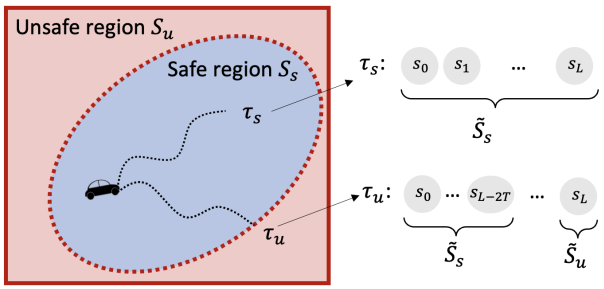


Fig. 3: Definitions of safe and unsafe sets. In safe demonstrations τ_s all state embeddings are labeled as safe. In contrast, in unsafe trajectories τ_u, only the first $(L - 2T)$ embeddings are treated as safe, where T is the Transformer context length, and only the last embedding is unsafe.

D. Online action optimization

We aim to learn a policy $\pi^* : s \to a$ that imitates good demonstrations while always being safe. At step t in testing, we compute π^* with a two-step approach shown in Fig. 2(b):
Action proposal. We feed sequence $s_{t-T+1}, a_{t-T+1}, ..., s_t$ to ConBaT and use the policy model π to get the action proposal $\hat{a}_t$, Then we propagate it through Transformer blocks to get the embedding $\hat{a}_t^+$ and use the future CBC to generate next state's safety score: $\hat{c}_{f,t+1} = C_f(s_t^+, \hat{a}_t^+)$.
Action optimization. If the future state violates the desired safety constraint, *i.e.* $\hat{c}_{f,t+1} < 0$, we optimize the action to keep s_{t+1}^+ within the safe set $\mathcal{S}_s^+$. We denote the new action $\hat{a}_t + \Delta a$ and solve for the following optimization problem:

$$\Delta a^* = \arg\min_{\Delta a} \left(\lambda |\Delta a|_2^2 + \max(-C_f(s_t^+, \hat{a}_t^+ + \Delta a), 0)\right) \tag{5}$$

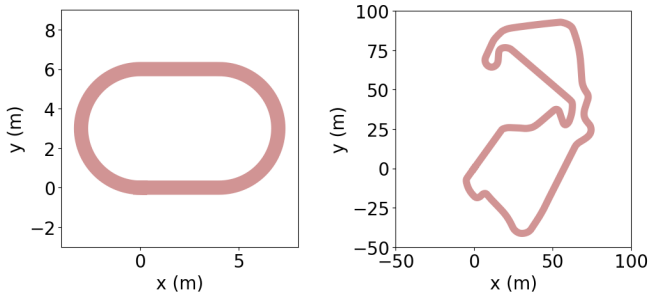
We use gradient descent to find the new action with the least deviation while ensuring safety. The agent takes a step $\hat{a}_t + \Delta a^*$, collects new observation, and the process repeats.

E. Training procedure

We train ConBaT in a two-phase paradigm. Phase I is analogous to the original pretraining scheme for PACT [12] which trains the policy head, world model, tokenizers and transformer blocks with the aim to learn reasonable agent behavior from demonstrations. For phase II, we freeze the base network weights, add control barrier critic modules and train with both the safe and unsafe demonstrations. By decoupling the training phases, we allow a user to potentially adapt the base policy π from a pretrained model towards different definitions of safety. Optionally, unsafe demonstrations can also be included in phase I to allow the world model to learn from the additional distribution of states, but the policy head is trained only on safe samples. We find that training the world model with unsafe demonstrations in phase I results in a better final performance.

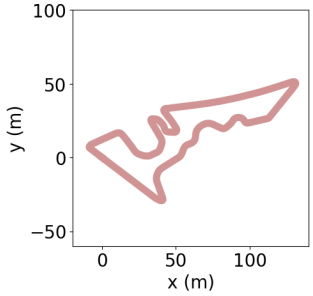
IV. Experimental Results

We consider two domains in simulation shown in Fig. 4:
F1/10 race car: A F1 car drives on racing tracks [43]. The (2-dim) observation is the distance and the angle relative to the center line. The action is the steering angle, and the goal



(a) F1/10 (playground)

(b) F1/10 (Silverstone)



(c) F1/10 (Austin)

(d) MuSHR env

Fig. 4: Simulation environment visualization.

is to drive safely without hitting track boundaries. **MuSHR navigation** [54]: A MuSHR car performs safe navigation in an indoor environment with 720-beam 2D LiDAR scans as observations and steering angles as actions. We use a 2D map scanned from a real office (30×70m) for data collection and deployment. We collect 1k trajectories from F1/10 and 10k trajectories from MuSHR ($\sim$ 75% safe and 25% unsafe), train ConBaT and evaluate its performance. We evaluate the learned policy by rolling out trajectories within the cut-off time horizon. We measure: (1) **collision rate**: the ratio of trajectories that crash before time-out; and (2) **average trajectory length (ATL)**: the average length of trajectories before crashing (within the cut-off horizon). Design details and additional results are in the supplementary video.

TABLE I: Comparison of PACT, PACT-FT and ConBaT for the F1/10 task on different tracks.

	Collision rate (%)			Avg. trajectory length		
	PACT	PACT-FT	ConBaT	PACT	PACT-FT	ConBaT
Playground	100	-	**0.0**	175.45	-	**1000**
Silverstone	100	96.88	**0.0**	61.57	439.28	**1000**
Austin	100	100	**61.7**	57.11	165.12	**678.14**

A. Safety analysis for F1/10 racing car on multiple tracks

We first compare ConBaT with PACT in the F1/10 task. We train models with 1K demonstrations (each has 100 timesteps) from the *Playground* track (Fig. 4a). In deployment we roll out 128 trajectories for 1000 timesteps. ConBaT gets 0% collision whereas PACT collides in every instance. Next, we apply the ConBaT trained only on *Playground* to more challenging tracks, *Silverstone* (Fig. 4b) and *Austin* (Fig. 4c). We compare against PACT trained only on *Playground* as well as a finetuned model on these tracks (PACT-FT). As shown in Table I, ConBaT can generalize to new tracks and outperform both PACT and PACT-FT. In Fig. 5, we show the histogram of the trajectory lengths on the test dataset on *Silverstone*. The PACT model trained on *Playground* registers fairly low ATL, whereas PACT-FT performs better by using more training samples, sometimes even reaches the max length. ConBaT significantly outperforms both, with all trajectories collision-free.

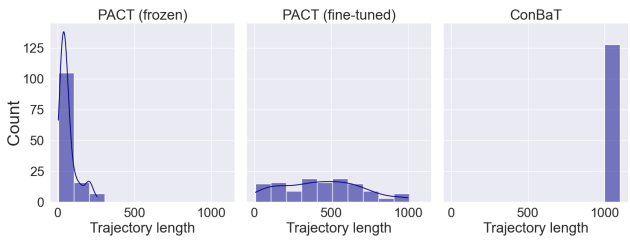


Fig. 5: Histogram of trajectory lengths on Silverstone track: ConBaT achieves a 100% safety, whereas both PACT and PACT-FT fail short.

B. Baselines comparison for MuSHR car navigation

In MuSHR we collect 10K demonstrations and train ConBaT and other baselines (if needed): Model predictive control (MPC), behavior cloning (BC), PACT, Generative Adversarial Imitation Learning (GAIL [31]), RL methods (PPO [53], TRPO [52], SAC [29]), constrained policy optimization (CPO) [1] and model-based control SABLAS [46]. In testing, we roll out 128 trajectories with 5000 timesteps. As shown in Fig. 6, even with the high-dimensional LiDAR inputs, ConBaT achieves the least collisions and the highest ATL. The inferior result of CPO and SABLAS could result from the extremely high dimensional observations ($720 \times 2 = 1440$) and complex constraints, whereas we learn the implicit dynamics and safety concept in the embedding space, which suits for complex dynamics and perception modalities.

C. Learning a new safety definition

ConBaT can quickly adapt to new safety concept beyond collision-avoidance without major architectural changes or hand-crafting cost functions. Consider a case where moving straight is undesirable and only curved motions are allowed (perhaps to create a dizzy rider experience in an amusement ride). We generate a new dataset with new safety labels and train a new critic ConBaT-NS (not-straight). Note that we don't finetune any PACT component - we expect the updated critic can choose "safe" behaviors. As seen in Fig. 7, ConBaT-NS can generate curved trajectories even in straight hallways, showing that new constraints can be incorporated with ease. Fig. 7c also shows how the new constraint is reflected in the policy distribution shift. Note that both ConBaT and ConBaT-NS are finetuned over the same PACT model, which demonstrates the potential for training multiple critics mapping to distinct safety constraints.

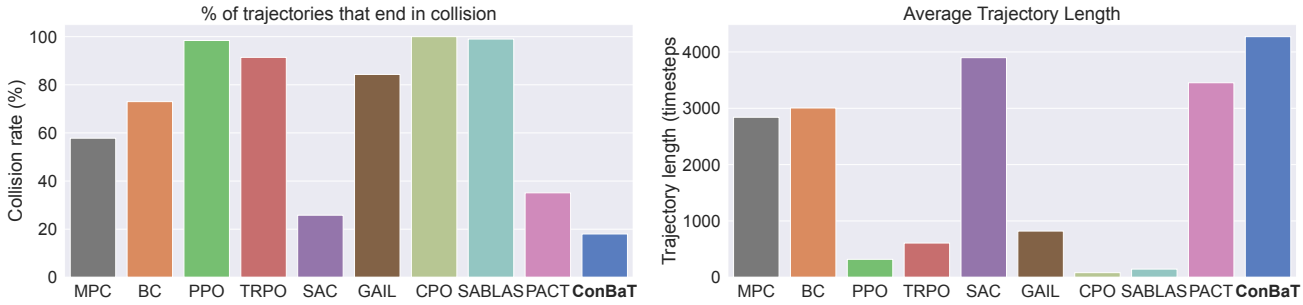
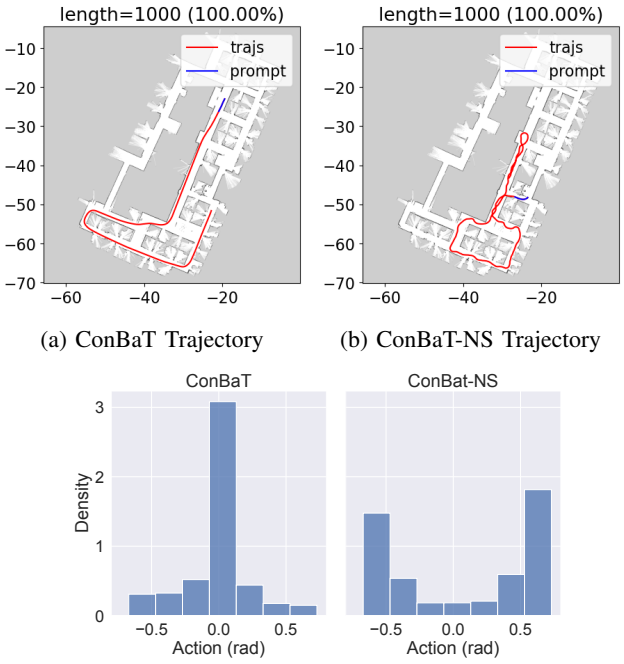


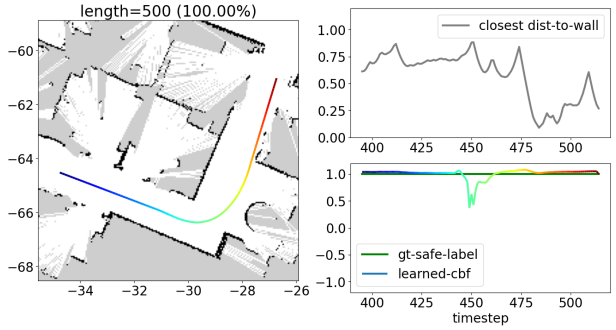
Fig. 6: ConBaT outperforms MPC and other learning-based methods on safe MuSHR navigation.



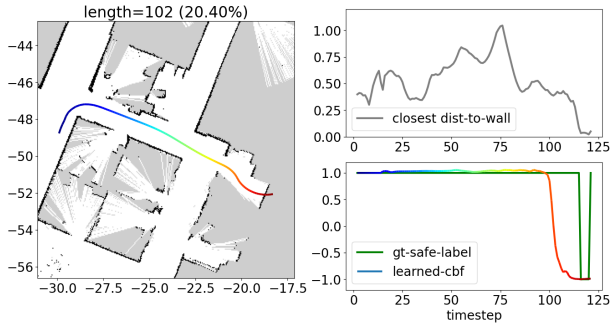
(a) ConBaT Trajectory (b) ConBaT-NS Trajectory

(c) Comparison of action distributions

Fig. 7: Demonstration of ConBaT-NS adapts to the new safety concept (driving straight is unsafe).



(a) Safe case: CBC>0. The CBC quantifies how hard to escape from a potential future collision (t=450).



(b) Unsafe case: CBC goes <0 as the agent enters irrecoverable states leading collision (120<t<125).

Fig. 8: Assessment on CBC values in MuSHR. The trajectory color indicates different timesteps.

D. CBC visualization

Fig. 8 assesses our learned CBC along typical trajectories in MuSHR simulation. We can see that the CBC value is positive for safe states and negative for unsafe states. Besides, a decreasing trend of CBC can be seen as the agent approaches a potential collision (100<t<120 in Fig. 8b).

E. Limitations

ConBaT cannot ensure safety if CBC makes wrong predictions due to out-of-distribution, or if the online optimization falls into local minima. One fix is to keep collecting new data (like DAGGER [49]) to update ConBaT. Besides, our runtime is $0.2 \sim 1.0$x higher than other learning-based methods due to online optimization. So far, we only train with vectorized inputs from simulations. In the future, we will extend to image inputs and more complicated real-world systems.

V. Conclusions

We propose ConBaT, a framework that learns safe and effective robot policies directly from positive and negative demonstrations. ConBaT leverages causal Transformers coupled with a safety critic (CBC) inspired by control barrier functions. The CBC implicitly builds a safe set for states from discrete safety labels, bypassing complex mathematical formulations for safety constraints. On two simulated domains we show that ConBaT outperforms existing classical and learning-based methods. Besides, ConBaT can quickly adapt to new safety concepts from limited demonstrations. This safety learning paradigm reduces training effort and makes it easier to adapt to diverse specifications.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[2] 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.

[3] 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.

[4] 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.

[5] Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pages 12–20, 1997.

[6] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129, 1995.

[7] Feryal Behbahani, Kyriacos Shiarlis, Xi Chen, Vitaly Kurin, Sudhanshu Kasewa, Ciprian Stirbu, Joao Gomes, Supratik Paul, Frans A Oliehoek, Joao Messias, et al. Learning from demonstration in the wild. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 775–781. IEEE, 2019.

[8] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

[9] Raunak P Bhattacharyya, Derek J Phillips, Changliu Liu, Jayesh K Gupta, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 789–795. IEEE, 2019.

[10] Raunak P Bhattacharyya, Derek J Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J Kochenderfer. Multi-agent imitation learning for driving simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1534–1539. IEEE, 2018.

[11] Aude Billard and Daniel Grollman. Robot learning by demonstration. *Scholarpedia*, 8(12):3824, 2013.

[12] Rogerio Bonatti, Sai Vemprala, Shuang Ma, Felipe Frujeri, Shuhang Chen, and Ashish Kapoor. Pact: Perception-action causal transformer for autoregressive robotics pre-training. *arXiv preprint arXiv:2209.11133*, 2022.

[13] Urs Borrmann, Li Wang, Aaron D Ames, and Magnus Egerstedt. Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine*, 48(27):68–73, 2015.

[14] Rafael Gomes Braga, Sina Karimi, Ulrich Dah-Achinanon, Ivanka Iordanova, and David St-Onge. Semantic navigation with domain knowledge. *arXiv preprint arXiv:2106.10220*, 2021.

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[16] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

[17] Arthur E Bryson and Yu-Chi Ho. *Applied optimal control: optimization, estimation, and control*. Routledge, 2018.

[18] Jyot Buch, Shih-Chi Liao, and Peter Seiler. Robust control barrier functions with sector-bounded uncertainties. *IEEE Control Systems Letters*, 6:1994–1999, 2021.

[19] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[20] Yuxiao Chen, Huei Peng, and Jessy Grizzle. Obstacle avoidance for low-speed autonomous vehicles with barrier function. *IEEE Transactions on Control Systems Technology*, 26(1):194–206, 2017.

[21] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3387–3395, 2019.

[22] Charles Dawson, Bethany Lowenkamp, Dylan Goff, and Chuchu Fan. Learning safe, generalizable perception-based hybrid control with certificates. *IEEE Robotics and Automation Letters*, 7(2):1904–1911, 2022.

[23] 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.

[24] Staffan Ekvall and Danica Kragic. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems*, 5(3):33, 2008.

[25] James Ferlez, Mahmoud Elnaggar, Yasser Shoukry, and Cody Fleming. Shieldnn: A provably safe nn filter for unsafe nn controllers. *arXiv preprint arXiv:2006.09564*, 2020.

[26] Jeremy H Gillula, Gabriel M Hoffmann, Haomiao Huang, Michael P Vitus, and Claire J Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *The International Journal of Robotics Research*, 30(3):335–354, 2011.

[27] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *2020 25th international conference on pattern recognition (ICPR)*, pages 10335–10342. IEEE, 2021.

[28] Luis Gressenbuch and Matthias Althoff. Predictive monitoring of traffic rules. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 915–922. IEEE, 2021.

[29] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[30] Sylvia Herbert, Jason J Choi, Suvansh Sanjeev, Marsalis Gibson, Koushil Sreenath, and Claire J Tomlin. Scalable learning of safety guarantees for autonomous systems using hamilton-jacobi reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5914–5920. IEEE, 2021.

[31] Jonathan Ho and Stefano Ermon. Generative adversarial imitation

learning. *Advances in neural information processing systems*, 29, 2016.

[32] Arne Kreutzmann, Diedrich Wolter, Frank Dylla, and Jae Hee Lee. Towards safe navigation by formalizing navigation rules. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 7(2):161–168, 2013.

[33] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *arxiv*, 2022.

[34] Xiao Li and Calin Belta. Temporal logic guided safe reinforcement learning using control barrier functions. *arXiv preprint arXiv:1903.09885*, 2019.

[35] Anqi Liu, Guanya Shi, Soon-Jo Chung, Anima Anandkumar, and Yisong Yue. Robust regression for safe exploration in control. In *Learning for Dynamics and Control*, pages 608–619. PMLR, 2020.

[36] Wenhao Luo, Wen Sun, and Ashish Kapoor. Sample-efficient safe learning for online nonlinear control with control barrier functions. *arXiv preprint arXiv:2207.14419*, 2022.

[37] Yue Meng, Zengyi Qin, and Chuchu Fan. Reactive and safe road user simulations using neural barrier certificates. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6299–6306. IEEE, 2021.

[38] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

[39] Sriraam Natarajan, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. Multi-agent inverse reinforcement learning. In *2010 ninth international conference on machine learning and applications*, pages 395–400. IEEE, 2010.

[40] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[41] Huansheng Ning, Rui Yin, Ata Ullah, and Feifei Shi. A survey on hybrid human-artificial intelligence for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[42] Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4209–4215. IEEE, 2021.

[43] Matthew O'Kelly, Hongrui Zheng, Dhruv Karthik, and Rahul Mangharam. F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. In *NeurIPS 2019 Competition and Demonstration Track*, pages 77–89. PMLR, 2020.

[44] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation*, pages 763–768. IEEE, 2009.

[45] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *International Workshop on Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.

[46] 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.

[47] 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.

[48] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[49] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[50] Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

[51] Jürgen Schmidhuber. Making the world differentiable: On using fully recurrent self-supervised neural networks for dynamic reinforcement learning and planning in non-stationary environments. *Institut für Informatik, Technische Universität München. Technical Report FKI-126*, 90, 1990.

[52] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[53] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[54] Siddhartha S. Srinivasa, Patrick Lancaster, Johan Michalove, Matt Schmittle, Colin Summers, Matthew Rockett, Joshua R. Smith, Sanjiban Chouhury, Christoforos Mavrogiannis, and Fereshteh Sadeghi. MuSHR: A low-cost, open-source robotic racecar for education and research. *CoRR*, abs/1908.08031, 2019.

[55] 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.

[56] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803. IEEE, 2010.

[57] Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. Safe reinforcement learning via curriculum induction. *Advances in Neural Information Processing Systems*, 33:12151–12162, 2020.

[58] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE international conference on robotics and automation*, pages 1928–1935. Ieee, 2008.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[60] Sai Vemprala and Ashish Kapoor. Adversarial attacks on optimization based planners. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9943–9949. IEEE, 2021.

[61] Li Wang, Dongkun Han, and Magnus Egerstedt. Permissive barrier certificates for safe stabilization using sum-of-squares. In *2018 annual American control conference (ACC)*, pages 585–590. IEEE, 2018.

[62] Peter Wieland and Frank Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467, 2007.

[63] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning. *arXiv preprint arXiv:2206.14176*, 2022.

[64] Dingjiang Zhou, Zijian Wang, Saptarshi Bandyopadhyay, and Mac Schwager. Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells. *IEEE Robotics and Automation Letters*, 2(2):1047–1054, 2017.