Safe and Performant Deployment of Autonomous Systems via Model Predictive Control and Hamilton-Jacobi Reachability Analysis

Hao Wang[1,2], Armand Jordana[3], Ludovic Righetti[3], Somil Bansal[2]

***Abstract*—While we have made significant algorithmic developments to enable autonomous systems to perform sophisticated tasks, it remains difficult for them to perform tasks effective and safely. Most existing approaches either fail to provide any safety assurances or substantially compromise task performance for safety. In this work, we develop a framework, based on model predictive control (MPC) and Hamilton-Jacobi (HJ) reachability, to optimize task performance for autonomous systems while respecting the safety constraints. Our framework guarantees recursive feasibility for the MPC controller, and it is scalable to high-dimensional systems. We demonstrate the effectiveness of our framework with two simulation studies using a 4D Dubins Car and a 6 Dof Kuka iiwa manipulator, and the experiments show that our framework significantly improves the safety constraints satisfaction of the systems over the baselines.**

I. Introduction

Recent advances in foundation models [23, 2] have transformed the field of robotics and rendered the prospect of robots performing useful tasks for us ever more realistic. In the field of robotics, large language models (LLM) and vision language models (VLM) have been employed to perform high level decision-making [20, 13] and synthesize action controls [15, 3], in diverse settings thanks to their generalizability.

Despite their impressive performance [22, 11], robot foundation models are prone to hallucinations and adversarial attacks [10, 14]. Since robots are designed to deploy around humans, their actions are often safety-critical, as safety violations can lead to irreversible damages. In this work, we develop a framework to safeguard autonomous systems, including foundation model-enabled robots, against pre-defined safety violations, while optimizing for the task performance of the robots.

Under the setting that we are given a set of constraints which the system should respect, a number of methods can be used to cooptimize the performance and safety of the robot. The most prominent approach in the data-driven robot control community is converting the safety constraints into penalties and incorporate them in the reward function [21, 7]. Though such approaches typically generate policies that encourage safe behaviors, they do not provide any safety guarantees. On the other hand, the control community poses the problem as a state-constrained optimal control problem and solve it using dynamic programming [4, 25]. This line of methods provides safety guarantees, but they cannot scale to high dimensional systems due to the curse of dimensionality associated with dynamic programming. Another popular family of methods is safety filtering [5, 24, 8], where the nominal controller is modified minimally if it is deemed unsafe. However, safety filtering is inherently myopic given its construction and typically leads to suboptimal task performance.

In this work, we propose to solve the problem of cooptimization using model predictive control (MPC) and Hamilton-Jacobi (HJ) reachability analysis. The core of our method is to incorporate the safety value function, obtained using HJ reachability analysis, as a final-time constraint in the MPC. Our method provides a closed-loop controller that optimizes the given performance objective while respecting pre-defined safety constraints, and our method can be efficiently scaled to high-dimensional systems and widely applicable to foundation model-enabled robots.

The contribution of this work is two-fold: 1) we propose a scalable framework for cooptimizing safety and performance for robotic systems, and 2) we demonstrate the effectiveness of our framework in two simulation studies with a 4D Dubins Car and 6 DOF Kuka iiwa manipulator.

II. Problem Formulation

We formulate the problem of cooptimization of safety and performance as a hierarchical control problem of the robot. The LLM/VLM, taking language commands from users, acts as the high-level symbolic planner, and it outputs a cost function encoding the desired behaviors. The MPC controller performs low-level trajectory optimization over the cost function and pre-defined safety constraints.

Formally, the low level trajectory optimization is formulated as a state-constrained optimal control problem [4] in Prob. 1. Let us use $\xi^{\mathbf{u}}_{x,t} : [t, T] \rightarrow \mathcal{X}$ to denote the state trajectory starting from state x at time t evolved with control signal $\mathbf{u} : [t, T) \rightarrow \mathcal{U}$. With a slight abuse of the notation, we use $\xi^{\mathbf{u}}_{x,t}(\tau)$ to denote the state at time $\tau \geq t$ along the trajectory $\xi^{\mathbf{u}}_{x,t}$.

[1] Author is with the Department of Electrical and Computer Engineering, University of Southern California. (email:{haowwang}@usc.edu)

[2] Authors are with the Department of Aeronautics and Astronautics, Stanford University. (email:{haowwang, somil@stanford.edu})

[3] Authors are with the Electrical and Computer Engineering Department, New York University. (email: {aj2988, ludovic.righetti}@nyu.edu)

This research is supported in part by the DARPA Assured Neuro Symbolic Learning and Reasoning (ANSR) program and by the NSF CAREER program (2240163).

Problem 1 (State-Constrained Optimal Control Problem)**.**

$$\inf_{\mathbf{u}} \quad J(x,t,\mathbf{u}) = \int_t^T r(\xi_{x,t}^{\mathbf{u}}(\tau), \mathbf{u}(\tau))d\tau + \phi(\xi_{x,t}^{\mathbf{u}}(T)) \tag{1a}$$

$$s.t. \quad \frac{d}{d\tau}\xi_{x,t}^{\mathbf{u}}(\tau) = f(\xi_{x,t}^{\mathbf{u}}(\tau), \mathbf{u}(\tau)) \ \forall \tau \in [t,T) \tag{1b}$$

$$l(\xi_{x,t}^{\mathbf{u}}(\tau)) \geq 0 \ \forall \tau \in [t,T] \tag{1c}$$

In this work, we are primarily interested in solving Prob. 1, as it optimizes for the desired behaviors commanded by the user, while respecting the pre-defined safety constraints. Crucially, the hierarchical structure and the set up of Prob. 1 ensures that the robot will not violate the pre-defined safety constraints despite possible hallucinations of or even adversarial attacks on the LLM/VLM.

III. Method

We solve Prob. 1 online in a model predictive control (MPC) fashion (i.e. solving Prob. 1 over a shorter planning horizon h, executing the first few controls, and solving Prob. 1 again from the evolved state), since the closed-loop nature of MPC provides robustness against potential modeling errors and unforeseen disturbances in deployment.

Though optimal control solvers have demonstrated impressive speed and reliability, in practice the planning horizon h of the MPC is much shorter than the task horizon T due to latency requirements. While it helps to reduce the online computation burden, the short planning horizon leads to myopic behaviors, often resulting in eventual violation of the safety constraints. The key novelty of our approach is utilizing the safety value function in the MPC formulation to ensure persistent satisfaction of the safety constraints.

A. MPC Formulation

In order to utilize the MPC framework to solve Prob. 1, we discretize the system dynamics and define the discrete-time state-constrained optimal control problem in Prob. 2. We denote the discrete-time dynamics, state trajectory, and control trajectory by f_d, $\mathbf{x}$ and $\mathbf{u}$, respectively.

Problem 2 (Discrete-Time State-Constrained Optimal Control Problem)**.**

$$\min_{\mathbf{u}} \quad J(x,\mathbf{u}) = \sum_{k=1}^{K} r(\mathbf{x}(k), \mathbf{u}(k)) + \phi(\mathbf{x}(K)) \tag{2a}$$

$$s.t. \quad \mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)) \ \forall k \in \{1,2,\ldots,K-1\} \tag{2b}$$

$$l(\mathbf{x}(k)) \geq 0 \ \forall k \in \{1,2,\ldots,K\} \tag{2c}$$

Suppose the planning horizon is given by h. Then at time step j, the MPC is formulated as follows.

Problem 3 (MPC)**.**

$$\min_{\mathbf{u}} \quad \sum_{k=j}^{j+h} r(\mathbf{x}(k), \mathbf{u}(k)) + \phi(\mathbf{x}(j+h)) \tag{3a}$$

$$s.t. \quad \mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)) \ \forall k \in \{j, j+1, \ldots, j+h-1\} \tag{3b}$$

$$l(\mathbf{x}(k)) \geq 0 \ \forall k \in \{j, j+1, \ldots, j+h\} \tag{3c}$$

B. Safety Value MPC

Due to the fact that the planning horizon is typically shorter than the task horizon, MPC is not guaranteed to be recursively feasible. When recursive feasibility does not hold, Prob. 3 becomes infeasible at some time step j, despite starting from a feasible state x. The most common approach to overcome this challenge is imposing a final-time constraint to ensure the system arrive at a set of states that are known to be recursively feasible. A desired final-time constraint has the following properties: 1) when satisfied, the system is guaranteed recursive feasibility, and 2) it should captures as "many" recursively feasible states as possible. In this subsection, we introduce how we construct the desired final-time constraint using Hamilton-Jacobi reachability analysis.

Given the state constraint in continuous-time Eq. (1c), we formulate the safety optimal control problem in Prob. 4.

Problem 4 (Safety Optimal Control Problem)**.**

$$\begin{aligned} \sup_{\mathbf{u}} \quad & J(x,t,\mathbf{u}) = \min_{\tau \in [t,T]} l(\xi_{x,t}^{\mathbf{u}}(\tau)) \\ s.t. \quad & \frac{d}{d\tau}\xi_{x,t}^{\mathbf{u}}(\tau) = f(\xi_{x,t}^{\mathbf{u}}(\tau), \mathbf{u}(\tau)) \ \forall \tau \in [t,T) \end{aligned} \tag{4}$$

The value function of Prob. 4, is given by

$$V_s(x,t) = \sup_{\mathbf{u}} \min_{\tau \in [t,T]} l(\xi_{x,t}^{\mathbf{u},\mathbf{d}}(\tau)) \tag{5}$$

In this work we consider the *converged* value function $V_s(x) = \lim_{t\to\infty} V_s(x,t)$, and we refer to this converged value function of Prob. 4 as the *safety value function*. It is important to note that the super 0-level set of the safety value function $\{x \in \mathcal{X} | V_s(x) \geq 0\}$ is the set of recursively feasible states considering the state constraint Eq. (1c) under the system dynamics Eq. (1b). Furthermore, the super 0-level set contains all possible recursively feasible states. The proofs for the aforementioned assertions can be found in [8].

Equipped with the safety value function, we incorporate it as a final-time constraint in the MPC formulation as follows. We implement the MPC controller using optimal control library Crocoddyl [18] and solver [12].

Problem 5 (Safety Value MPC)**.**

$$\min_{\mathbf{u}} \quad \sum_{k=j}^{j+h} r(\mathbf{x}(k), \mathbf{u}(k)) + \phi(\mathbf{x}(j+h)) \tag{6a}$$

$$s.t. \quad \mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)) \ \forall k \in \{j, j+1, \ldots, j+h-1\} \tag{6b}$$

$$l(\mathbf{x}(k)) \geq 0 \ \forall k \in \{j, j+1, \ldots, j+h-1\} \tag{6c}$$

$$V_s(\mathbf{x}(j+h)) \geq 0 \tag{6d}$$

IV. Experiments

A. 4D Dubins Car

In this case study, we simulate a 4D Dubins car with the following dynamics

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ u_1 \\ u_2 \end{bmatrix}$$

where x, y, θ, and v are the $x-$position, $y-$position, heading, and velocity of the system. u_1 and u_2 are the controls of the system. The system is tasked with moving from a random initial state to a goal state, defined in the $x-y$ plane, while avoiding obstacles over a 2 seconds time horizon. More formally, the running cost and the terminal cost, defined in Eq. (6b), are given by $r(x) = \phi(x) = ||[x, y]^\top - [x_g, y_g]^\top||_2$, where $[x_g, y_g]^\top$ is the goal state. The obstacles and the goal are shown in Figure. 1.

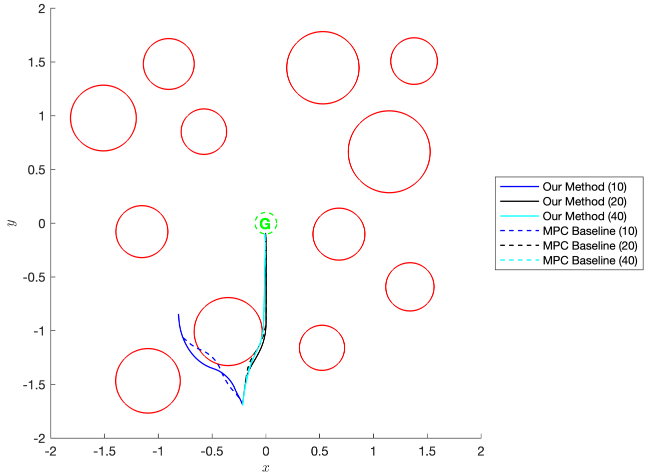

Fig. 1. Dubins4D obstacle configuration and rollouts for our method and the baselines

We use a task horizon of 2 seconds and MPC time step of 0.01 second. We consider a baseline MPC controller, which is identical to our method except for not employing the safety value function as a final time constraint in Eq. (6d). We also vary the planning horizon (10, 20, and 40 steps) to demonstrate its effect on the safety and performance of the system. In this case study, the safety value function is computed using the LevelSetToolBox [19] and HelperOC [1] over a $50 \times 50 \times 50 \times 30$ grid. While we synthesize the MPC controllers online, we compute the safety value function offline.

For evaluation, we focus on the rollout success rate and the cost of the synthesized state trajectories. For clarity, we compare the results to our method with 20 planning steps. As shown in TABLE I, our method is consistently safe (with the exception of 1 trial) regardless of the planning horizon. On the other hand, the rollout success rate increases with the planning horizon. This is expected since with longer planning horizon, the MPC is able to reason about the safety constraint further into the future, and hence reducing the number of safety violations. Our method essentially condenses all the reasoning regarding the safety constraint into the safety value function, and as a result, it satisfies the safety constraint given any planning horizon.

Consistent satisfaction of the safety constraints comes at a slight cost of the task performance. As we can see in TABLE I, our method with 20 planning steps is only able to obtain better task performance on 45.98% of the trials than the MPC baseline with the same planning horizon. This is also expected because we employ the *converged* safety value function in our formulation, and it typically leads to slight more conservative behaviors. Also unsurprisingly, the task performance of our method and the MPC baseline improves as the planning horizon increases.

TABLE I
Comparison of metrics for our method and the baselines

	Rollout Success Rate	% Trajectories with Higher Cost Compared to Our Method (20)
Our Method (10)	100%	93.62%
Our Method (20)	100%	-
Our Method (40)	99%	9.68%
MPC Baseline (10)	80%	89.47%
MPC Baseline (20)	90%	45.98%
MPC Baseline (40)	98%	6.45%

B. 6 DoF Kuka iiwa Manipulator

Consider a scenario where the manipulator is tasked with moving a cup of coffee quickly through a cluttered environment without spilling. Though motion planning can enable the manipulator to perform agile obstacle avoidance maneuvers, it cannot do so while considering the dynamic effects of the manipulator on the coffee, likely leading to spills or running into the obstacles. In this case study, we aim to synthesize the dynamic, agile, and safe behaviors that are required to perform the prescribed task. We simulate a 6 DOF Kuka iiwa manipulator with a 12D state space (6 joint position variables and 6 joint velocity variables) and 6D control space (6 joint torques). Note that the wrist joint is locked since it is irrelevant for this experiment. The task of the manipulator is moving from the initial state to the goal end-effector position in the operation space, while avoiding obstacles in the operation space and keeping the joint accelerations within certain intervals. The joint acceleration constraints are in placed to prevent the manipulator from moving too rapidly and spilling the coffee.

Let us denote the joint position, velocity, and acceleration by q, v and a, and we use the shorthand $FK(\cdot)$ for forward kinematics, mapping joint positions to (x, y, z) positions of the end-effector. We define the failure set to be a cylindrical region in the operation space, and the end-effector of the manipulator should avoid entering into this failure set. More concretely, the running cost and final cost are given by $r(q, v, u) = \phi(q, v, u) = ||FK(q) - x_g||_2$, the obstacle constraint is given by $l(FW(q)) \geq 0$, and the joint acceleration constraints are given by $\underline{a_i} \leq a_i \leq \overline{a_i}\ \forall i \in \{1, \ldots, 6\}$, where x_g is the goal end-effector position, $\underline{a_i}$ and $\overline{a_i}$ are the lower and upper bounds for the i^{th} joint acceleration.

Given the dimensionality of the system, we use a learning-based approach [6, 9] to compute the safety value function. For this experiment, we use task horizon of 1 second, planning horizon of 15 steps, and MPC time step of 0.01 second. We again consider the MPC baseline where the safety value function constraint is not imposed. We roll out our method and the MPC baseline from 15 initial states, and our method succeeds in 11 out of 15 trials, while the MPC baseline only succeeds in 3 out of 15 trials. The end-effector trajectories of both methods for one of the trials are shown in Fig. 2. Qualitatively, our method is more cautious around the obstacle, as it moves away from the obstacle before turning towards the goal. On the other hand, the MPC baseline heads directly for the goal and inevitably arrives at a state where it cannot avoid entering into the failure set.

It is important to note that the MPC baseline incorporates the obstacle constraint Eq. (1c). However, due to the limited planning horizon, it is unable to reason about the long-term safety and eventually leads the system into a state where it cannot satisfy the obstacle constraint.

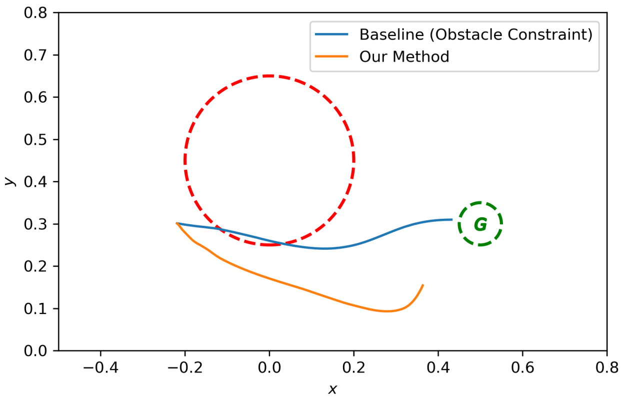

Fig. 2. End-effector trajectories of our method and the MPC baseline for one of the trials

V. Conclusion

In this work, we propose a scalable framework that cooptimizes safety and performance for autonomous systems, including foundation model-enabled robots. Our method incorporates the safety value function as a final-time constraint in the MPC formulation, and it is shown to improve the safety constraint satisfaction of the system in simulation studies. However, our method replies on learning-based methods for computing the safety value function for high-dimensional systems, and as a result, our framework currently cannot provide formal safety guarantees. We look to address this shortcoming in future works by utilizing verification techniques [16, 17].

References

[1] helperOC Library, 2019. https://github.com/HJReachability/helperOC.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[4] Albert Altarovici, Olivier Bokanowski, and Hasnaa Zidani. A general Hamilton-Jacobi framework for nonlinear state-constrained control problems. *ESAIM: COCV*, 19(2):337–357, 2013. doi: 10.1051/cocv/2012011. URL https://doi.org/10.1051/cocv/2012011.

[5] 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.

[6] Somil Bansal and Claire J Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2021.

[7] Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. *arXiv preprint arXiv:2010.14497*, 2020.

[8] Javier Borquez, Kaustav Chakraborty, Hao Wang, and Somil Bansal. On safety and liveness filtering using hamilton-jacobi reachability analysis. *IEEE Transactions on Robotics*, 2024.

[9] Zeyuan Feng, Le Qiu, and Somil Bansal. Bridging model predictive control and deep learning for scalable reachability analysis. *arXiv preprint arXiv:2505.03830*, 2025.

[10] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[11] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[12] Armand Jordana, Sébastien Kleff, Avadesh Meduri, Justin Carpentier, Nicolas Mansard, and Ludovic

Righetti. Stagewise implementations of sequential quadratic programming for model-predictive control.

[13] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B Murthy. Position: Llms can't plan, but can help planning in llm-modulo frameworks. In *Forty-first International Conference on Machine Learning*, 2024.

[14] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

[15] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[16] Albert Lin and Somil Bansal. Generating formal safety assurances for high-dimensional reachability. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10525–10531. IEEE, 2023.

[17] Albert Lin and Somil Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. In *6th Annual Learning for Dynamics & Control Conference*, pages 719–731. PMLR, 2024.

[18] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hammoud, Maximilien Naveau, Justin Carpentier, Ludovic Righetti, Sethu Vijayakumar, and Nicolas Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2536–2542. IEEE, 2020.

[19] Ian M Mitchell et al. A toolbox of level set methods. *UBC Department of Computer Science Technical Report TR-2007-11*, page 31, 2007.

[20] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

[21] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020.

[22] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[23] 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.

[24] Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

[25] Hao Wang, Adityaya Dhande, and Somil Bansal. Cooptimizing Safety and Performance With a Control-Constrained Formulation. *IEEE Control Systems Letters*, 2024.