Safely Integrating Perception, Planning, and Control for Robust Learning-Based Robot Autonomy

Glen Chou
Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105

Many of the most impactful applications of robotics (e.g., autonomous driving, assisted living, robotic surgery, etc.) are *safety-critical*; that is, robots in these domains must be able to reliably complete human-specified tasks while obeying hard safety constraints. Traditional motion planners [25] can achieve this if the robot state is perfectly known, the environment is accurately modeled, and the task is unambiguous. However, in reality, we need our robots to operate with noisy sensors in unstructured environments where tasks may be temporally-extended and vaguely-specified. To succeed in these difficult scenarios, robots must rely on data to refine their understanding of their environments and tasks. This often takes the form of learned models in the autonomy stack, e.g., learned perception systems [21], dynamics models [17], task specifications [19], and controllers [26]. However, since data is limited, these learned models are often inaccurate and unreliable, and blindly trusting them can cause catastrophe (e.g., fatal self-driving car crashes [1, 32]). To avoid such tragedies, we must verify that behaviors planned by the robot will safely complete the task. However, not all task representations and models are easily amenable to safety verification, and existing techniques have yet to scale to analyze modern deep-learning-based models within the autonomy loop, which has interconnections that cause errors to propagate and accumulate.

My goal is to close this gap so that robots can use learned models across all levels of the autonomy stack while guaranteeing end-to-end safety and robustness. More specifically, my research is motivated by the following two questions. First, how can robots learn task representations that can *define and enable* safe behavior (i.e., hard constraints)? Second, how can robots *use* the learned tasks and models in a way that remains robust to uncertainty and error, guaranteeing safety and task completion *for the entire robot autonomy pipeline at runtime*?

I. Completed Work

To begin to answer these questions, I will go over my work so far on 1) learning task constraints from demonstrations, and on using the uncertainty in the learned task to compute safer plans (Sec. I-A), and 2) certifying safety and robustness with learned perception modules, dynamics, and controllers in the autonomy loop (Sec. I-B). See Fig. 1 for a map of my work.

A. Learning constrained task specifications

1) Overview: Task specification is a critically-understudied aspect of safe autonomy – without an unambiguous description of the set of truly safe, task-completing behaviors, a notion of

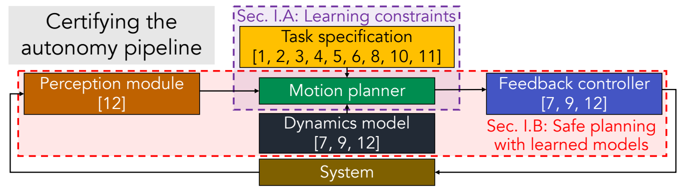

Fig. 1. This diagram summarizes my work on certifying the end-to-end safety and robustness of a robot autonomy pipeline with learned components.

safety and robustness cannot even be defined. To this end, I design algorithms that guarantee safety and task completion when planning with uncertain, learned task specifications.

Consider a robot with state x, control u, and known dynamics $x_{t+1} = f(x_t, u_t)$. To find a plan $\xi = (x_1, u_1, x_2, \dots)$ that safely completes a task, that task must be concretely specified, often as a (time-dependent) set of constraints $\mathbf{g}_t(x_t, u_t) \geq 0$. However, it is difficult for humans to specify $\mathbf{g}_t(\cdot, \cdot)$ explicitly; it is more natural to describe the task through demonstrations (e.g., physically [2] or via virtual reality [35]). Then, to safely generalize to similar tasks, the robot must learn the task constraints $\mathbf{g}_t(\cdot, \cdot)$ that are implicit in the demonstrations.

This problem is closely related to inverse optimal control (IOC) [22]. IOC assumes that the demonstrator solves $\min_\xi c(\xi)$, where the cost function $c(\xi)$ encodes the task, and aims to learn $c(\xi)$ from optimal solutions. The cost $c(\xi)$ softens the hard task constraints (e.g., it penalizes collisions instead of avoiding them altogether), which can lead to unsafe behavior when planning with the learned cost function [12].

2) Methods: To instead enforce safety by construction, I develop novel algorithms that explicitly learn the hard task constraints $\mathbf{g}_t(\cdot, \cdot)$. In contrast to IOC, I assume the demonstrator solves a *constrained* problem, $\min_{\mathbf{g}(\xi) \geq 0} c(\xi)$. The crux of this problem relies on learning what not to do (i.e., unsafe behavior) *using only safe examples* – we do not want unsafe behavior to be executed just to simplify the learning. I achieve this via the core insight that the demonstrations' approximate optimality (e.g., following the shortest or minimum energy feasible trajectory to the goal) *implicitly* defines what the robot should not do, i.e., trajectories that achieve a lower cost than the demonstration must violate the unknown task constraint. After learning, our method can generate new safe plans by optimizing trajectories subject to the learned constraint.

Assuming approximately *globally-optimal* demonstrations, I develop a sampling-based method for learning unknown, *time-independent* task constraints [7, 8, 12]. I relax this assumption in [10, 16] via the Karush-Kuhn-Tucker (KKT) conditions

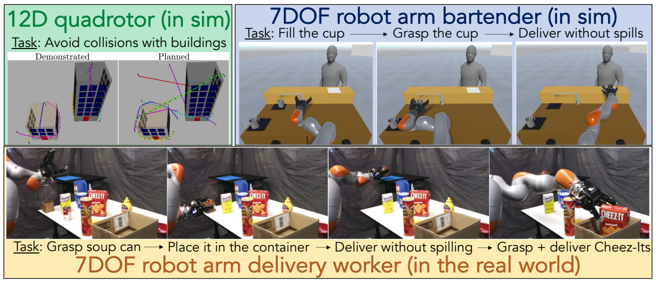

Fig. 2. Using the methods in Sec. I-A, we learn these constrained tasks: (Top left) quadrotor collision avoidance (in sim); (Top right) multi-step Kuka bartender task (in sim); (Bottom) multi-step Kuka delivery task (real world).

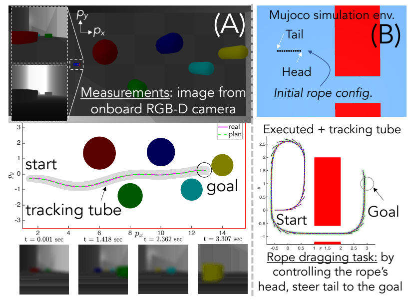

Fig. 3. Planning safely with learned models (Sec. I-B). (A): nonholonomic car safely stabilizing around a plan via RGB-D image-based feedback control. Top: environment; middle: executed/planned trajectory; bottom: runtime observations. (B): 22D rope: safe steering to the goal via state feedback.

[6] – a notion of *local demonstrator optimality*; thus, after learning the task constraints, the robot can improve relative to the demonstrator. I have also explored suboptimality due to partial information [23]. I have explored grid [7, 12], parametric [8, 10], and nonparametric [16] constraint representations; each trades off expressiveness, scalability, and data-efficiency. In [11, 14], I extend these ideas to learn *time-dependent* constraints in the form of *linear temporal logic (LTL) formulas*, which can represent complex, multi-stage tasks. Finally, in [9], I tackle the issue of non-uniqueness – in general, many constraints can explain the demonstrations. My previous work tries to satisfy all possible constraints in planning, but this is overly conservative. To address this, [9] explicitly quantifies this constraint uncertainty by forming a *belief over constraints* and updating it with online data, iteratively computing probabilistically-safe plans that enable eventual task completion.

3) Results: My methods learn a broad class of manipulation and navigation tasks from demonstrations, including quadrotor obstacle avoidance, a multi-step 7DOF arm robot bartending task, and a multi-step 7DOF arm object delivery task (Fig. 2). Moreover, planning with these learned constraints leads to safe task completion at runtime.

B. Planning safely with learned models

1) Overview: Here, I seek to guarantee safety through the *combined* perception, planning, and control pipeline, when the perception module, dynamics, and controller are learned and represented as neural networks. The core challenges here lie in 1) bounding how the error in the learned models accumulates and degrades the tracking ability of the learned controller, and 2) how to use this bound together with the learned models to plan towards some goal while guaranteeing that the true system can be safely stabilized around this plan at runtime.

This problem is by no means entirely new. Robust control [37, 31] and feedback motion planning [34, 28, 33] can bound the effect of model error on trajectory tracking, but require *a priori* known error bounds which are difficult to obtain when the model is a neural network. Other work in safe learning-based control generally requires perfect state feedback [30, 5], *a priori* known stabilizing controllers [3], assumed accuracy of model error bounds [20, 4], or full state-invertibility from observations [18]. To my knowledge, *my method is the first to certify safety for a learning-enabled autonomy pipeline from rich observations (e.g., images) down to low-level control.*

2) Methods: My core insight is that to guarantee safety in the face of unreliable learned dynamics and perception modules, the method must infer *where* the models can be trusted, and *to what extent*. In [24, 13, 15], my method first bounds the model error in a domain around the training data by estimating its Lipschitz constant; I call this the "trusted domain". My method propagates these error bounds (for both the dynamics and perception) through a state estimator and feedback controller based on contraction theory [29, 27] to derive a certified tracking error bound. This bound can guide a planner to remain in low-error parts of the "trusted domain", leading to plans that can be safely tracked when using RGB-D images and a learned perception module in the control loop.

3) Results: I demonstrate these methods in simulation on a variety of systems, including a nonholonomic car (Fig. 3.A), demonstrating that my method safely and reliably steers the system to the goal using RGB-D sensor measurements. Using state feedback, I also demonstrate that my method can solve a highly-underactuated 22D deformable object manipulation task (Fig. 3.B). In all of these examples, my method results in safe behavior at runtime, whereas baseline approaches that exit the "trusted domain" are often unsafe.

II. Future Work

Moving forward, I am interested in combining our constraint-learning (Sec. I-A) and safe learning-based planning methods (Sec. I-B) to guarantee safety using unreliable perception models, dynamics, controllers, *and task specifications*. Combining these methods can also enable the learning of task constraints from rich observations under uncertain dynamics.

I also believe that my safe learning-based planning methods (Sec. I-B) can be improved via *adaptation*. For instance, by leveraging online data, we may be able to tighten the tracking error bounds, enabling the execution of more aggressive maneuvers. Adaptation can also be achieved via replanning; extending to this setting requires carefully considering recursive feasibility and a more computationally-efficient planner.

Finally, I want to extend my work on safe perception-based control (Sec. I-B) to handle *real images*. The key challenge lies in accurately bounding the perception error from limited real data; this can be achieved by quantifying the sim-to-real gap [36] and bounding its effect on the perception system.

REFERENCES

[1] Evan Ackerman. Fatal tesla self-driving car crash reminds us that robots aren't perfect, Jun 2021. URL https://spectrum.ieee.org/fatal-tesla-autopilot-crash-reminds-us-that-robots-arent-perfect.

[2] Baris Akgün, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz. Trajectories and keyframes for kinesthetic teaching: a human-robot interaction perspective. In *HRI*, pages 391–398. ACM, 2012.

[3] Felix Berkenkamp, Riccardo Moriconi, Angela P Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *CDC*, pages 4661–4666, 2016.

[4] Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 908–918, 2017.

[5] Nicholas M. Boffi, Stephen Tu, Nikolai Matni, Jean-Jacques E. Slotine, and Vikas Sindhwani. Learning stability certificates from data. *CoRL*, 2020.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[7] Glen Chou, Dmitry Berenson, and Necmiye Ozay. Learning constraints from demonstrations. *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

[8] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning parametric constraints in high dimensions from demonstrations. In *Conference on Robot Learning (CoRL)*, 2019.

[9] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Uncertainty-aware constraint learning for adaptive safe motion planning from demonstrations. In *Conference on Robot Learning (CoRL)*, 2020.

[10] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning constraints from locally-optimal demonstrations under cost function uncertainty. *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[11] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Explaining multi-stage tasks by learning temporal logic formulas from suboptimal demonstrations. In *Robotics: Science and Systems (RSS)*, 2020.

[12] Glen Chou, Dmitry Berenson, and Necmiye Ozay. Learning constraints from demonstrations with grid and parametric representations. *International Journal of Robotics Research (IJRR)*, 2021.

[13] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Model error propagation via learned contraction metrics for safe feedback motion planning of unknown systems. *IEEE Conference on Decision and Control (CDC)*, 2021.

[14] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning temporal logic formulas from suboptimal demonstrations: theory and experiments. *Autonomous Robots (AuRo)*, 2022.

[15] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Safe output feedback motion planning from images via learned perception modules and contraction theory. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2022.

[16] Glen Chou, Hao Wang, and Dmitry Berenson. Gaussian process constraint learning for scalable chance-constrained motion planning from demonstrations. *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[17] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *NeurIPS*, pages 4759–4770, 2018.

[18] Sarah Dean, Andrew J. Taylor, Ryan K. Cosner, Benjamin Recht, and Aaron D. Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning (CoRL)*, volume 155 of *Proceedings of Machine Learning Research*, pages 654–670. PMLR, 2020.

[19] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 49–58, 2016.

[20] Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy H. Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. Autom. Control.*, 64(7):2737–2752, 2019.

[21] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Robotics: Science and Systems*, 2018.

[22] R. E. Kalman. When is a linear control system optimal? *Journal of Basic Engineering*, 86(1):51–60, Mar 1964.

[23] Craig Knuth, Glen Chou, Necmiye Ozay, and Dmitry Berenson. Inferring obstacles and path validity from visibility-constrained demonstrations. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2020.

[24] Craig Knuth, Glen Chou, Necmiye Ozay, and Dmitry Berenson. Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants. *IEEE Robotics and Automation Letters (RA-L)*, 2021.

[25] Steven LaValle. *Planning algorithms*. Cambridge university press, 2006.

[26] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17:39:1–39:40, 2016.

[27] Winfried Lohmiller and Jean-Jacques E. Slotine. On contraction analysis for non-linear systems. *Autom.*, 34 (6):683–696, 1998.

[28] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *IJRR*, 36 (8):947–982, 2017.

[29] Ian R. Manchester and Jean-Jacques E. Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Trans. Autom. Control.*, 62(6):3046–3053, 2017.

[30] Gaurav Manek and J. Zico Kolter. Learning stable deep dynamics models. In *NeurIPS*, pages 11126–11134, 2019.

[31] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *TAC*, 50(7):947–957, 2005.

[32] David Shepardson. U.s. identifies 12th tesla autopilot car crash involving emergency vehicle, Sep 2021. URL https://www.reuters.com/business/autos-transportation/us-identifies-12th-tesla-assisted-systems-car-crash-involving-emergency-vehicle-2021-09-01/.

[33] Sumeet Singh, Benoit Landry, Anirudha Majumdar, Jean-Jacques E. Slotine, and Marco Pavone. Robust feedback motion planning via contraction theory. 2019.

[34] Russ Tedrake. Lqr-trees: Feedback motion planning on sparse randomized trees. *Robotics: Science and Systems V*, 2009.

[35] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *ICRA*, pages 1–8. IEEE, 2018.

[36] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI*, pages 737–744. IEEE, 2020.

[37] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*. 1998.