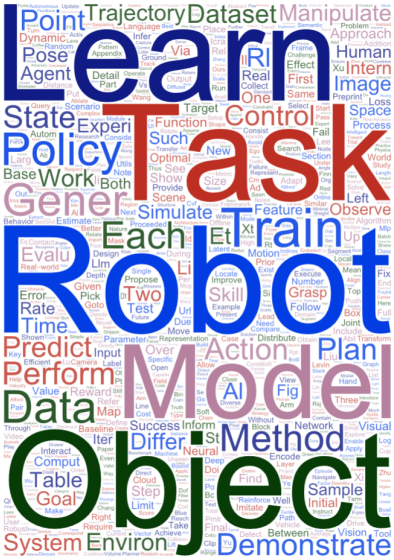


Robot Learning

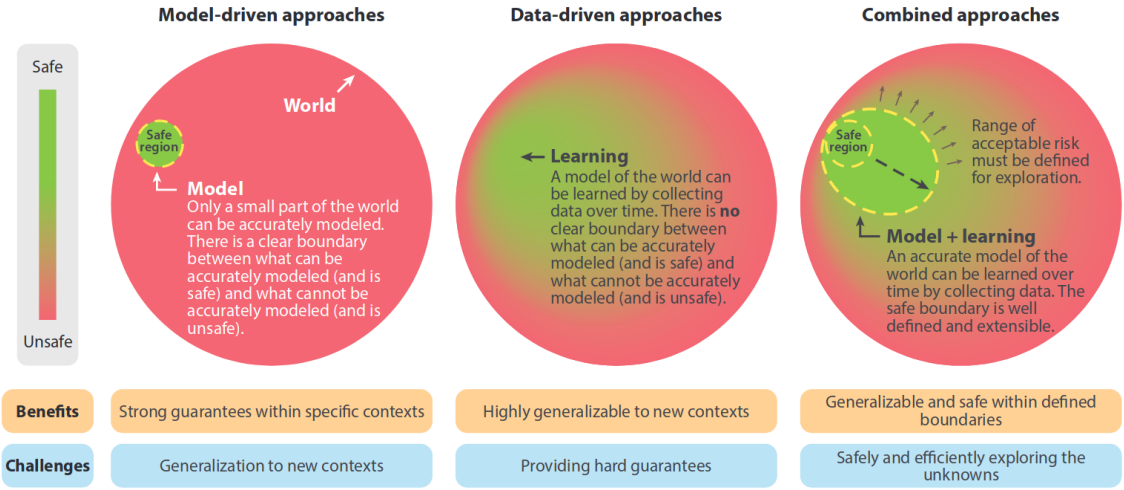
Safe Learning

Wolfgang Hönig
Technical University of Berlin
Summer 2024

Safety

What might “safety” refer to in safe learning?

Motivation

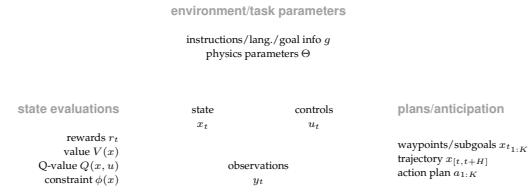


L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. 5:411–444.
URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211, doi:10.1146/annurev-control-042920-020211

Outline

- Definitions of Safety and Safe Learning
- Overview of Existing Solutions (& Case Studies)
- Discussion / Open Challenges

What is learned?



- Consider policy $\pi : x_t \mapsto u_t$
  - Safety means (intuitively) that if we rollout π ($x_{t+1} = f(x_t, \pi(x_t)) \quad \forall t$), we never end up in a "bad" state (e.g., collision, crash, stability/tracking) for "valid" start states x_0
  - In some cases, safety should apply while learning as well

Definition of Safety (1)

- Dynamics $x_{k+1} = f_k(x_k, u_k, w_k)$
  - $x_k \in \mathcal{X}$ (state)
  - $u_k \in \mathcal{U}$ (action)
  - $w_k \sim \mathcal{W}$ (process noise)

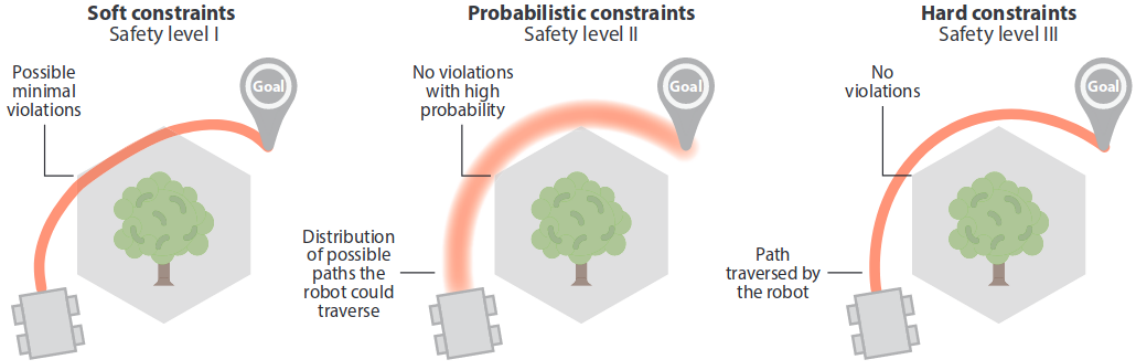
Definition of Safety (1)

- Dynamics $x_{k+1} = f_k(x_k, u_k, w_k)$
  - $x_k \in \mathcal{X}$ (state)
  - $u_k \in \mathcal{U}$ (action)
  - $w_k \sim \mathcal{W}$ (process noise)
  - Why f_k and not f?

Definition of Safety (1)

- Dynamics $x_{k+1} = f_k(x_k, u_k, w_k)$
  - $x_k \in \mathcal{X}$ (state)
  - $u_k \in \mathcal{U}$ (action)
  - $w_k \sim \mathcal{W}$ (process noise)
  - Why f_k and not f?
- Objective $J(x_{0:N}, u_{0:N-1}) = l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, u_k)$
- Safety constraints
  - State constraints (e.g., no collisions)
  - Input constraints (e.g., actuation limits)
  - Stability guarantees (e.g., robot converging to desired reference path)

Definition of Safety (2)



L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. 5:411–444.
URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211, doi:10.1146/annurev-control-042920-020211

Definition of Safety (3)

- Hard constraints (safety level 3)

$$c_k^j(x_k, u_k, w_k) \leq 0 \quad \forall k \quad \forall j$$

- Chance constraints (safety level 2)

$$Pr(c_k^j(x_k, u_k, w_k) \leq 0) \geq p^j \quad \forall k \quad \forall j \quad p^j \in [0, 1]$$

- Soft constraints (safety level 1)

$$c_k^j(x_k, u_k, w_k) \leq \epsilon_j \quad \forall k \quad \forall j$$

$$l_\epsilon(\boldsymbol{\epsilon}) \geq 0 \text{ (Cost function term)}$$

Definition of Safe (Control) Learning

Safe Robot Control Problem

$$\min_{\boldsymbol{\pi}_{0:N-1}, \boldsymbol{\epsilon}} \quad J(\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}) + l_\epsilon(\boldsymbol{\epsilon})$$

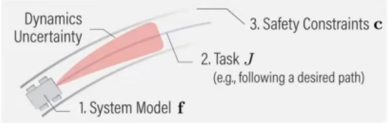
$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k),\ \mathbf{w}_k \sim \mathcal{W},\ \forall k \in \{0, ..., N-1\},$$

hard, probablistic, or soft safety constraints $\mathbf{c}$,

$$\mathbf{x}_0 = \bar{\mathbf{x}}_0,$$

$$\mathbf{u}_k = \boldsymbol{\pi}_k(\mathbf{x}_k)$$

Each component may be unknown or partially known!



Safe Learning Control (SLC) Design

$$\text{SLC} : (\mathcal{P}, \mathcal{D}) \mapsto \boldsymbol{\pi}_k$$

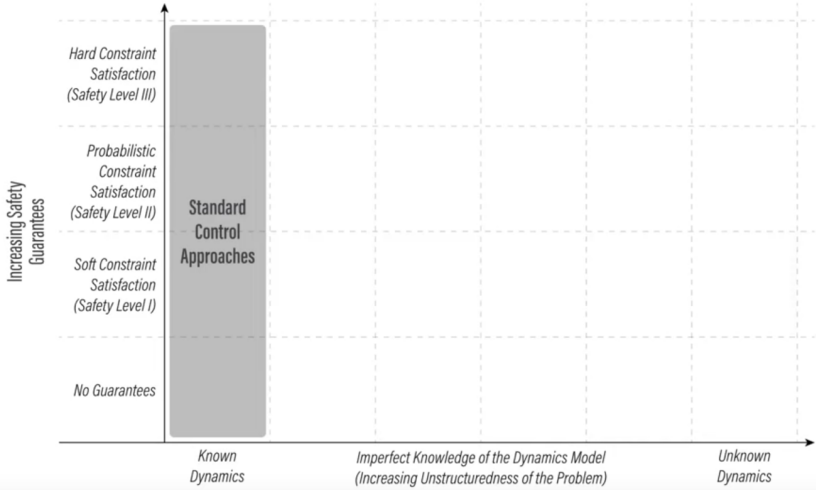
Prior Knowledge $\mathcal{P} = \{\bar{J}, \bar{\mathbf{f}}, \bar{\mathbf{c}}\}$

Data $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{u}^{(i)}, \mathbf{c}^{(i)}, l^{(i)}\}_{i=0}^{i=D}$

Relationship to (Classic) Controls

- Robust control
  - Assume disturbance bounds known
  - Find *fixed* controller that works even in the worst-case
- Adaptive controls
  - Assume environment has *varying* parameters Θ (not directly observed)
  - Controller changes *online* (e.g., by estimating Θ)
- Tube-based Model Predictive Control (MPC)
  - Robust control in MPC framework: use tighter constraints to account for unmodeled dynamics

Relationship to (Classic) Controls

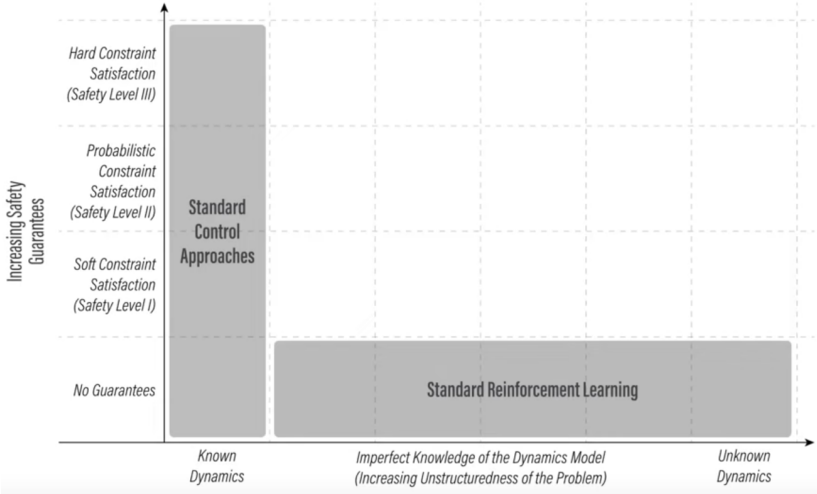


Control Approach

prior model (possibly with bounded uncertainties)

- MPC
- adaptive control
- robust control
- ...

Relationship to (Classic) RL



Control Approach

prior model (possibly with bounded uncertainties)

- MPC
- adaptive control
- robust control
- ...

data collected from interacting with the environment

RL Approach

Outline

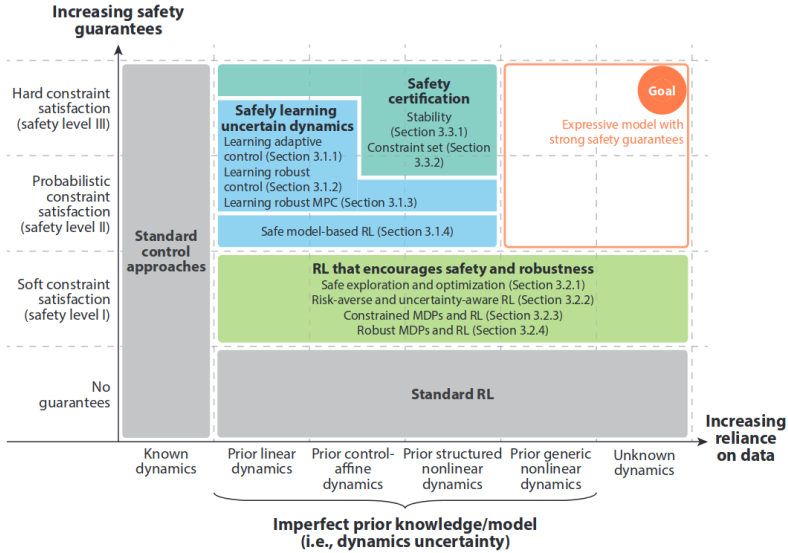
- Definitions of Safety and Safe Learning
- **Overview of Existing Solutions (& Case Studies)**
- Discussion / Open Challenges

Existing Solution Strategies

(i) Safely Learn Uncertain Dynamics
(ii) RL that Encourages Safety and Robustness
(iii) Safety Certification

[Online Adaption/Learning (dynamics, cost function, constraints, control parameters) vs Offline (update in batches)]

Existing Solution Strategies



L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. 5:411–444. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211, doi:10.1146/annurev-control-042920-020211

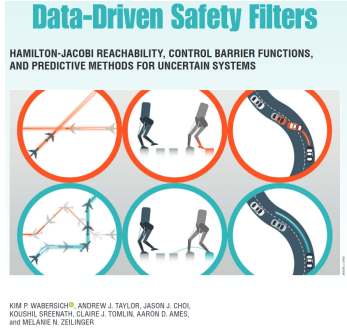
Strategy III: Safety Certification: Constraint Set

- Key idea
  - Learn policy “as usual”
  - At runtime, apply a safe action $u_{\text{safe}} = \text{argmin}_u \|u - u_{\text{learned}}\|^2$ such that x_{k+1} is safe
- Safe states can be computed by
  - Control Barrier Functions (CBFs)
  - Hamilton-Jacobi Reachability Analysis
  - Predictive safety filters

    [keep track of safe control inputs that could steer back to a known safe state]

Strategy III: Safety Certification: Constraint Set

- More Advanced
  - If safety layer is differentiable $\rightarrow$ end-to-end training (e.g. [7])
  - Learn safety filters directly



K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger. Data-Driven Safety Filters: Hamilton-Jacobi Reachability, Control Barrier Functions, and Predictive Methods for Uncertain Systems.
43(5):137–177.
URL: https://ieeexplore.ieee.org/document/10266799/, doi:10.1109/MCS.2023.3291885

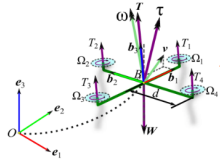
Strategy III: Safety Certification: Stability

- Stability: (informal) Can the robot track the reference, even with (small) disturbances? [Formal proofs via Lyapanov functions or contraction theory]
- Typical assumptions:
  - Bounded disturbance
  - Bounded change in disturbance (Lipschitz continuous with known Lipschitz bound)
  - Unbounded control authority
- Lipschitz-based: Treat neural network as “disturbance”; limit magnitude and Lipschitz bound during training (*Spectral Normalization*) (e.g., [8])
- Region of Attraction: Lyapunov Neural Networks [6]

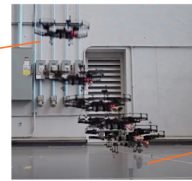
Case Study: Neural Lander (based on slides from Shi)

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u + f(q,\dot{q},u)$$

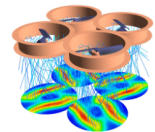
- $f(q,\dot{q},u)$ is the unknown aerodynamics depending on u
- Idea: use a DNN $\hat{f}(q,\dot{q},u)$ to approximate $f(q,\dot{q},u)$
- *Q: How to guarantee stability?*



easy to model and symbolic



Neural-Lander
[Shi et al., ICRA'19]



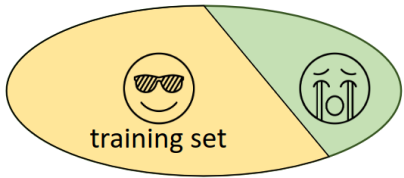
much harder to model!

Video: `https://youtu.be/FLLsG0S78ik`

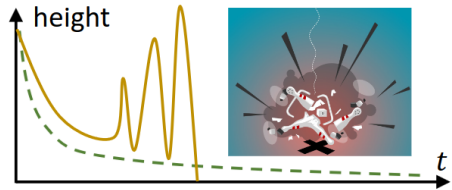
Case Study: Neural Lander (based on slides from Shi)

- Do we have to constraint the DNN? Yes! If we don't:

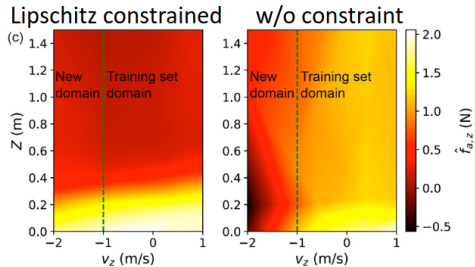
Learning perspective: $\hat{f}$ can not generalize



Control perspective: closed-loop instability



2D heatmaps of the learned $\hat{f}$



Strategy II: RL that Encourages Safety and Robustness

- 1. Safe Exploration and Optimization
- 2. Risk-averse RL and uncertainty-aware RL
- 3. RL for Constrained MDPs (CMDPs)
- 4. RL for Robust MDPs

Strategy II: RL that Encourages Safety: Safe Exploration

- Safe Exploration: only allow the policy to explore safe states

Safe Exploration in Markov Decision Processes

Teodor Mihai Moldovan MOLDOVAN@CS.BERKELEY.EDU
Pieter Abbeel PABBEEL@CS.BERKELEY.EDU
University of California at Berkeley, CA 94720-1758, USA

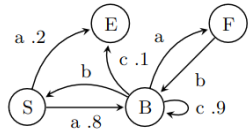


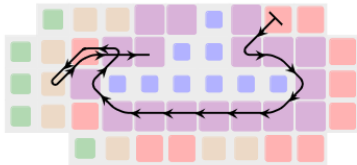
Figure 1. Starting from state S, the policy (aababab...) is safe at a safety level of .8. However, the policy (acccc...) is not safe since it will end up in the sink state E with probability 1. State-action Sa and state B can neither be considered safe nor unsafe, since both policies use them.

T. M. Moldovan and P. Abbeel. Safe exploration in Markov decision processes.
In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1451–1458. Omnipress

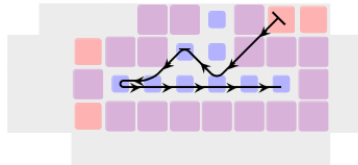
Strategy II: RL that Encourages Safety: Safe Exploration

- Safe Exploration: only allow the policy to explore safe states

(a) Based on the available information after the first step, moving South-West is unsafe.



(b) The safe explorer successfully uncovers all of the map by avoiding irreversible actions.

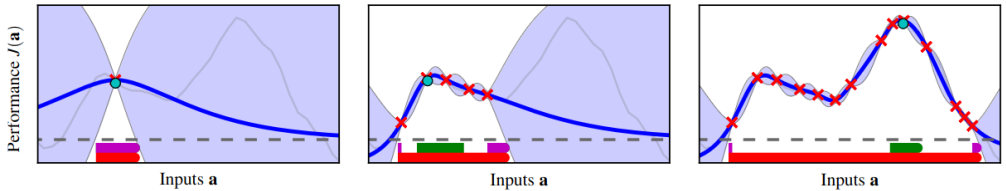


(c) The adapted R-MAX explorer gets stuck before observing the entire map.

T. M. Moldovan and P. Abbeel. Safe exploration in Markov decision processes.
In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1451–1458. Omnipress

Strategy II: RL that Encourages Safety: Safe Exploration

- Safe Optimization: Minimize cost function without sampling inputs that violate safety constraints, e.g., SafeOpt [1]



(a) Initial, safe parameters. (b) After 5 evaluations: local maximum found. (c) After 13 evaluations: global maximum found.

Safe set $\mathcal{S}_n$ (red): Could be potential maximizers $\mathcal{M}_n$ (green) or expanders $\mathcal{G}_n$ (magenta)

Case Study: SafeOpt

Algorithm 1: Modified SafeOpt algorithm

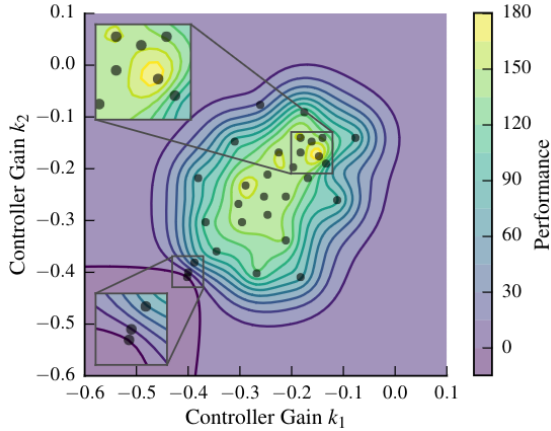
Inputs: Domain $\mathcal{A}$
Safe threshold $J_{\min}$
GP prior $(k(\mathbf{a}_i, \mathbf{a}_j),\ \sigma_\omega^2)$
Initial, safe controller parameters $\mathbf{a}_0$

Initialize GP with $(\mathbf{a}_0,\ \hat{J}(\mathbf{a}_0))$
for $n = 1, \dots$ **do**
  $\mathcal{S}_n \leftarrow \{\mathbf{a} \in \mathcal{A} \mid l_n \geq J_{\min}\}$
  $\mathcal{M}_n \leftarrow \{\mathbf{a} \in \mathcal{S}_n \mid u_n(\mathbf{a}) \geq \max_{\mathbf{a}'} l_n(\mathbf{a}')\}$
  $\mathcal{G}_n \leftarrow \{\mathbf{a} \in \mathcal{S}_n \mid g_n(\mathbf{a}) > 0\}$
  $\mathbf{a}_n \leftarrow \operatorname{argmax}_{a \in \mathcal{G}_n \cup \mathcal{M}_n} w_n(\mathbf{a})$
  Obtain measurement $\hat{J}(\mathbf{a}_n) \leftarrow J(\mathbf{a}_n) + \omega_n$
  Update GP with $(\mathbf{a}_n,\ \hat{J}(\mathbf{a}_n))$
end

- Update sets using GPs
- From the union of safe potential maximizers or expanders, measure where the uncertainty is highest

Case Study: SafeOpt

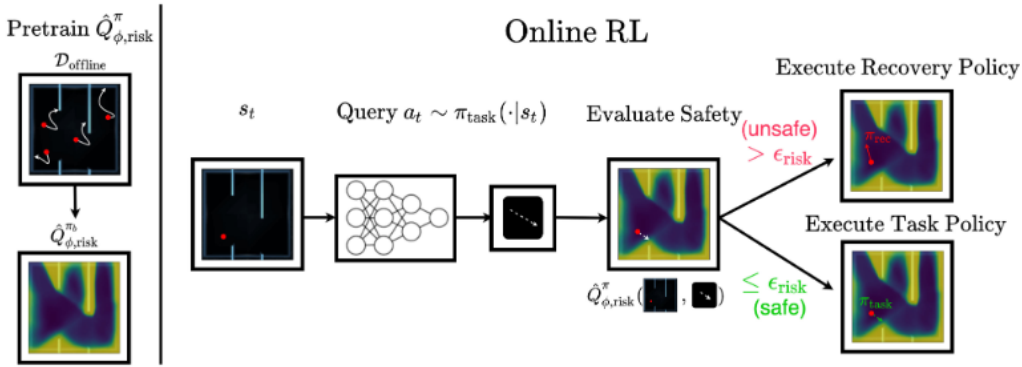
Application: Safe controller gain tuning



Video: `https://youtu.be/GiqNQdzc5TI`

Strategy II: RL that Encourages Safety: Safe Exploration

- Learning a safety critic: learn a Q-function that predicts “safety”, e.g., [9]

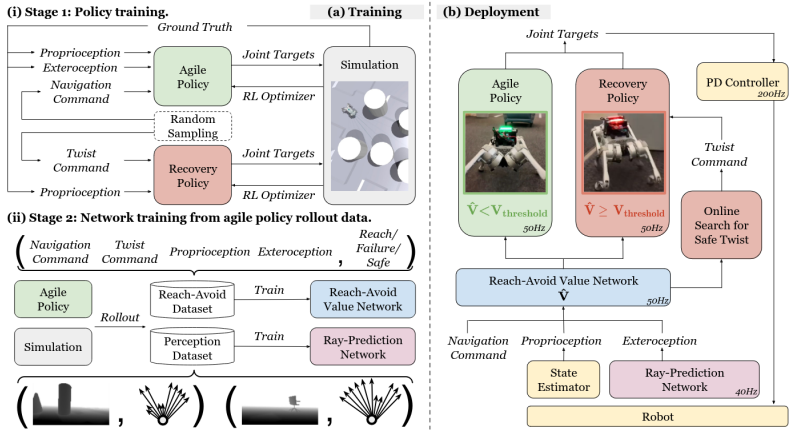


Recovery RL: For intuition, we illustrate Recovery RL on a 2D maze navigation task where a constraint violation corresponds to hitting a wall. I

Strategy II: RL that Encourages Safety: Risk-averse RL

- Learn/estimate *risks* (e.g., probability of a collision)
- At runtime, prefer actions with low risk (e.g., MPC planner)

Case Study: Agile But Safe [3]



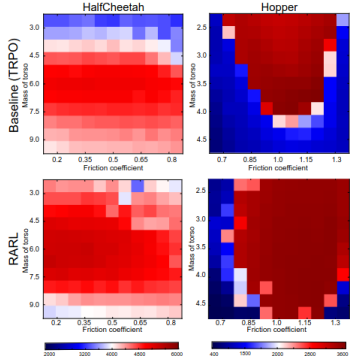
Web: `https://agile-but-safe.github.io/`

Strategy II: RL that Encourages Safety: RL for CMDPs

> “However, most of the work in this area remains confined to naive simulated tasks, motivating further research on their applicability in real-world control.”

Strategy II: RL that Encourages Safety: RL for Robust MDPs

- Robust Adversarial RL [5]



- Train two policies: a robust policy and a destabilizing adversary (that can apply random forces on the robot)
- Trained iteratively

- Domain Randomization

Strategy I: Safely Learn Uncertain Dynamics

- 1. Learning Adapative Control
- 2. Learning Robust Control
- 3. Learning Robust MPC
- 4. Safe Model-based RL

Outline

- Definitions of Safety and Safe Learning
- Overview of Existing Solutions (& Case Studies)
- **Discussion / Open Challenges**

Open Challenges

- Broader class of robots (hybrid dynamics, multi-robot, soft-robot, ...)
- Scalability & Sampling/Computational Efficiency
- Imperfect State Measurements
- Verification of Safety-Related Assumptions
- Automatic Inference about What is Safe

Discussion

- What about other learning problems?
  - Learning planners that output waypoints/trajectories (rather than a policy that outputs one action)?
  - Using humans as input (e.g., through language)?
  - Including perception (e.g., $y \mapsto u$)
  - We discussed Safe RL and safe dynamics learning; What would Safe Imitation Learning be? What would Safe Inverse RL be?
- How would you safely learn how to fly from scratch?

Conclusion

- Three Safety Levels: soft constraints, chance constraints, hard constraints
- Safety filters can be easily used, but are difficult to design for uncertain dynamics
- Encouraging safety has other advantages (e.g., sim-to-real transfer)
- Many practical challenges remain, especially for full robotic solutions

[1] F. Berkenkamp, A. P. Schoellig, and A. Krause.
Safe controller optimization for quadrotors with Gaussian processes.
In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 491–496. IEEE.
URL: http://ieeexplore.ieee.org/document/7487170/, doi:10.1109/ICRA.2016.7487170.

[2] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig.
Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning.
5:411–444.
URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211,
doi:10.1146/annurev-control-042920-020211.

[3] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi.
Agile But Safe: Learning Collision-Free High-Speed Legged Locomotion.
URL: https://arxiv.org/abs/2401.17583v3.

[4] T. M. Moldovan and P. Abbeel.
Safe exploration in Markov decision processes.
In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1451–1458. Omnipress.

[5] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta.
Robust Adversarial Reinforcement Learning.
In *Proceedings of the 34th International Conference on Machine Learning*, pages 2817–2826. PMLR.
URL: https://proceedings.mlr.press/v70/pinto17a.html.

[6] S. M. Richards, F. Berkenkamp, and A. Krause.
The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamical Systems.
URL: http://arxiv.org/abs/1808.00924, arXiv:1808.00924, doi:10.48550/arXiv.1808.00924.

[7] B. Riviere, W. Honig, Y. Yue, and S.-J. Chung.

GLAS: Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning With End-to-End Learning.
5(3):4249–4256.
URL: https://ieeexplore.ieee.org/document/9091314/, doi:10.1109/LRA.2020.2994035.

[8] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung.
Neural Lander: Stable Drone Landing Control Using Learned Dynamics.
In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE.
URL: https://ieeexplore.ieee.org/document/8794351/, doi:10.1109/ICRA.2019.8794351.

[9] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg.
Recovery RL: Safe Reinforcement Learning With Learned Recovery Zones.
6(3):4915–4922.
URL: https://ieeexplore.ieee.org/document/9392290/, doi:10.1109/LRA.2021.3070252.

[10] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger.
Data-Driven Safety Filters: Hamilton-Jacobi Reachability, Control Barrier Functions, and Predictive Methods for Uncertain Systems.
43(5):137–177.
URL: https://ieeexplore.ieee.org/document/10266799/, doi:10.1109/MCS.2023.3291885.