Learning for Safety-Critical Control in Dynamical Systems

Yisong Yue
Policy/Controller Learning (Reinforcement & Imitation)

**Goal:** Find “Optimal” Policy

**Imitation Learning:**
Optimize imitation loss

**Reinforcement Learning:**
Optimize environmental reward

Learning-based Approach for Sequential Decision Making

Non-learning approaches include: optimal control, robust control, adaptive control, etc.
Imitation Learning Tutorial

https://sites.google.com/view/icml2018-imitation-learning/

Yisong Yue
yyue@caltech.edu
@YisongYue
visongyue.com

Hoang M. Le
Hoang.Le@Microsoft.com
@HoangMinhLe
hoangle.info
Many Exciting Success Stories

![Diagram of AlphaFold and Microsoft Azure Personalizer](image-url)
“I want to use deep learning to optimize the design, manufacturing and operation of our aircrafts. But I need some guarantees.” -- Aerospace Director
Behavioral Guarantees

- Fairness
- Low-risk
- Temporal logic
- Etc...

Stability

Safety

Smoothness

\[ \frac{dx}{dt}, V(x_1, x_2) \]

\[ g(x, k) \]

\[ f(x, k) \]

\[ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \]

\[ \begin{pmatrix} dX \\ dt \end{pmatrix} \]

\[ \begin{pmatrix} \text{grad } V \\ \text{grad } V \end{pmatrix} \]

\[ V(x_1, x_2) \]

\[ x \]

\[ \text{Safe Set} \]

\[ \text{Ideal Behavior} \]

\[ \text{Unsmooth} \]

\[ \text{Smooth Recovery} \]
Research Questions

• How to constrain learning to (provably) satisfy guarantees?

• How to integrate domain knowledge from physics & control theory?
  • (Towards) a unified framework?

• How to exploit structure for faster learning?
  • (both computational & statistical)
Integration of Learning at Varying Levels

- Integration in control/action
- Integration in dynamics modeling
- Integration in optimization problem
Starting Point

In general, very hard to verify/optimize!

\[
\begin{align*}
\arg\min_h & \ L(h) \\
\text{s.t.} & \ R(h) < \kappa
\end{align*}
\]

(e.g., for all inputs, h is safe)

- Model-Based/Free
- On/Off Policy
- Imitation/Reinforcement
- Optimal Control
**Functional Regularization**
(to a certified controller)

\[
\arg\min_h L(h) \\
\text{s.t.} \\
\exists g \in G: \|h - g\|^2 < \kappa
\]

---

**Model-Based Controllers**
(certified by construction)

Key idea: G encodes domain knowledge & guarantees
Blended Policy Class (solution concept)

\[ \arg\min_{h=(f,g)} L(h) \quad \text{s.t.} \quad h(s) = \arg\min_{a'} (f(s) - a')^2 + \lambda (g(s) - a')^2 = \frac{f(s) + \lambda g(s)}{1 + \lambda} \]

Smooth Imitation Learning for Online Sequence Prediction, Hoang Le, Andrew Kang, Yisong Yue, Peter Carr. ICML 2016
Control Regularization for Reduced Variance Reinforcement Learning, Richard Cheng, Abhinav Verma, et al. ICML 2019
Test-Time Functional Regularization

Certified Complex Predictors $H$

Complex Predictors $F$

$$\argmin_{h=(f,g)} L(h) \quad \text{s.t.} \quad h(s) = \arg\min_{a'} (f(s) - a')^2 + \lambda (g(s) - a')^2 = \frac{f(s) + \lambda g(s)}{1 + \lambda}$$

Smooth Imitation Learning for Online Sequence Prediction
Hoang Le, Andrew Kang, Yisong Yue, Peter Carr. ICML 2016
Theoretical Guarantees

\[ \arg \min_{h=(f,g)} L(h) \quad \text{s.t.} \quad h(s) = \arg \min_{a'} (f(s) - a')^2 + \lambda (g(s) - a')^2 \]

\[ = \frac{f(s) + \lambda g(s)}{1 + \lambda} \]

• By construction: h “close” to g
  • Certifications on g => (relaxed) certifications on h

• Compatible with IL/RL
  • New learning approaches

• Very data efficient

Run-time regularization

Convergence analysis

Low-Variance Gradients
Comments on Certified by Construction

$$h(s) = \frac{f(s) + \lambda g(s)}{1 + \lambda}$$

• **Assumption:** all $g \in G$ are certified by construction
  • Robust against disturbances
  • Satisfied for many physical systems

• **Disturbance:** Black box predictor $f \in F$ is a “disturbance” of $g$
  • Worst-case disturbance depends $\max_{s} f(s)$ and $\lambda$
  • Guarantees worsen as $\lambda$ decreases

• **Note:** local guarantee at the per-state level
Comments on Optimization/Learning

\[
\argmin_{h=(f,g)} L(h) \quad \text{s.t.} \quad h(s) = \argmin_{a'} (f(s) - a')^2 + \lambda (g(s) - a')^2 \\
= \frac{f(s) + \lambda g(s)}{1 + \lambda}
\]

- Alternating optimization
  - Hold \( g \) fixed, optimize \( f \)
  - Hold \( h \) fixed, optimize \( g \)
  - (see NeurIPS 2019 paper for clean treatment)

*Reduces to “standard” approaches*

Imitation-Projected Programmatic Reinforcement Learning
Abhinav Verma, Hoang Le, Yisong Yue, Swarat Chaudhuri. NeurIPS 2019
Naïve Approach

- Supervised learning of demonstration data
  - Train predictor per frame
  - Predict per frame

In practice, 2-step smoothing:
Smooth Policy Class

\[
\begin{align*}
\text{argmin}_{h=(f,g)} L(h) \quad &\text{s.t.} \quad h(s) = \text{argmin}_{a'} (f(s) - a')^2 + \lambda (g(s) - a')^2 \\
&= \frac{f(s) + \lambda g(s)}{1 + \lambda}
\end{align*}
\]
Test-Time Functional Regularization

\[ \arg\min_{h=(f,g)} L(h) \quad \text{s.t.} \quad h(s) = \arg\min_{a'} (f(s) - a')^2 + \lambda (g(s) - a')^2 \]

\[ = \frac{f(s) + \lambda g(s)}{1 + \lambda} \]
Our Results

Smooth Imitation Learning for Online Sequence Prediction
Hoang Le, Andrew Kang, Yisong Yue, Peter Carr. ICML 2016
Qualitative Comparison

Our Approach

Learning Online Smooth Predictors for Real-time Camera Planning using Recurrent Decision Trees

Jianhui Chen, Hoang Le, Peter Carr, Yisong Yue, Jim Little. CVPR 2016
Control Regularization

\[ h(s) = \frac{f(s) + \lambda g(s)}{1 + \lambda} \]

• f is black box
• g is “control prior” (e.g., H-infinity controller)
• Learn f using any RL method

Control Regularization for Reduced Variance Reinforcement Learning
Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, Joel Burdick. ICML 2019
Control Regularization

• (Relaxed) Lyapunov stability bounds:

Control Regularization for Reduced Variance Reinforcement Learning
Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, Joel Burdick. ICML 2019
Control Regularization

\[ h(s) = \frac{f(s) + \lambda g(s)}{1 + \lambda} \]

• Theorem (informal):
  • Variance of policy gradient decreases by factor of: \( \left( \frac{1}{1+\lambda} \right)^2 \)
  • Bias converges to: \( \left( \frac{\lambda}{1+\lambda} \right) D_{TV}(h^*, g) \)

Implies much faster learning!

Control Regularization for Reduced Variance Reinforcement Learning
Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, Joel Burdick. ICML 2019
Control Regularization

Figure 4. Learning results for CartPole, Car-Following, and TORCS RaceCar Problems. (a) Reward improvement over control prior using DDPG with different set values for $\lambda$ or an adaptive $\lambda$. The right plot is a zoomed-in version of the left plot without variance bars for clarity. Values above the dashed black line signify improvements over the control prior. (b) Performance and variance in the reward as a function of the regularization, across different runs of the algorithm using random initializations/seeds. Dashed lines show the performance (i.e. reward) and variance using the adaptive weighting strategy. Variance is measured for all episodes across all runs. Adaptive and intermediate values of $\lambda$ exhibit best learning. Again, performance is baselined to the control prior, so any performance value above 0 denotes improvement over the control prior.

It is important to note that using the adaptive strategy for setting $\lambda$ in the TORCS setting gives us the highest-performance policy that improves upon the control prior. The variance with the adaptive strategy is significantly lower than for the DDPG baseline, which again shows that the learning process reliably learns a good controller.

7. Conclusion

This paper shows, through theoretical results and experimental validation, that our method of functional regularization with a control prior enables significant variance reduction and performance improvements in reinforcement learning. This regularization can be interpreted as constraining the explored action space during learning. Our method also allows us to capture dynamic stability properties of a robust control prior to guarantee stability during learning. A significant criticism of RL is that random initializations/seeds can produce vastly different learning behaviors, limiting application of RL to physical systems. Our framework substantially alleviates this problem, allowing reliable learning of high-performance, stable controllers with minimal variability.
Summary: Functional Regularization

Regularization ↔ Constrained Learning

$$\arg\min_h L(h) \quad s.t. \quad R(h) < \kappa$$

IL/RL Objective

$$\arg\min_h L(h) + \lambda R(h)$$

Side Guarantees

Hybrid Policy Solution Concept

$$h(s) = \arg\min_{a'} (f(s) - a')^2 + \lambda(g(s) - a')^2$$

$$= \frac{f(s) + \lambda g(s)}{1 + \lambda}$$
Summary: Functional Regularization (cont.)

• Control methods => analytic guarantees (side guarantees)

• Blend w/ learning => improve precision/flexibility (real-world improvements)

• Preserve side guarantees (possibly relaxed)

• Interpret as functional regularization (speeds up learning)

• Other directions:

  Batch Policy Learning under Constraints
  Hoang Le, Cameron Voloshin, Yisong Yue. ICML 2019 (offline learning)

  Imitation-Projected Programmatic Reinforcement Learning
  Abhinav Verma, Hoang Le, Yisong Yue, Swarat Chaudhuri. NeurIPS 2019 (programmatic controllers)
Integration of Learning at Varying Levels

• Integration in control/action

• Integration in dynamics modeling

• Integration in optimization problem
Model-Based Control

\[ s_{t+1} = F(s_t, u_t) + \epsilon \]

- New State
- Current State
- Current Action (aka control input)
- Unmodeled Disturbance / Error

Robust/Optimal Control (fancy contraction mappings)
- Stability guarantees (e.g., Lyapunov)
- Precision/optimality depends on error

(Value Iteration is also contraction mapping)
Learning Residual Dynamics

\[ s_{t+1} = F(s_t, u_t) + \tilde{F}(s_t, u_t) + \epsilon(s_t, u_t) \]

\( F \) = nominal dynamics
\( \tilde{F} \) = learned dynamics

Leverage robust/optimal control (fancy contraction mappings)
- Preserve stability (even using deep learning)
- Requires \( \tilde{F} \) Lipschitz & bounded error
Stable Drone Landing

Ground effect

Neural Lander: Stable Drone Landing Control using Learned Dynamics
Guanya Shi, Xichen Shi, Michael O’Connell, Rose Yu, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, Soon-Jo Chung. ICRA 2019
Control System Formulation

- Dynamics:

\[
\begin{align*}
\dot{p} &= v, \\
\dot{v} &= mg + Rf_u + f_a \\
\dot{R} &= RS(\omega), \\
J\dot{\omega} &= J\omega \times \omega + \tau_u + \tau_a
\end{align*}
\]

- Control:

\[
\begin{align*}
f_u &= [0, 0, T]^\top \\
\tau_u &= [\tau_x, \tau_y, \tau_z]^\top
\end{align*}
\]

\[
\begin{bmatrix}
T \\
\tau_x \\
\tau_y \\
\tau_z
\end{bmatrix} =
\begin{bmatrix}
cT & cT & cT & -cT \\
0 & cTL_{arm} & 0 & -cTL_{arm} \\
-cTL_{arm} & 0 & cTL_{arm} & 0 \\
-cQ & cQ & -cQ & cQ
\end{bmatrix}
\begin{bmatrix}
n_1^2 \\
n_2^2 \\
n_3^2 \\
n_4^2
\end{bmatrix}
\]

- Unknown forces & moments:

\[
\begin{align*}
f_a &= [f_{a,x}, f_{a,y}, f_{a,z}]^\top \\
\tau_a &= [\tau_{a,x}, \tau_{a,y}, \tau_{a,z}]^\top
\end{align*}
\]
Data Collection (Manual Exploration)

- Learn ground effect: $\tilde{F}(s, u) \rightarrow f_a = [f_{a,x}, f_{a,y}, f_{a,z}]^T$
- $(s,u)$: height, velocity, attitude and four control inputs

Notable Extension: Safe Exploration

Ensures $\tilde{F}$ is Lipshitz
[Bartlett et al., NeurIPS 2017]
[Miyato et al., ICLR 2018]

Spectral-Normalized 4-Layer Feed-Forward
Prediction Results

Neural Lander: Stable Drone Landing Control using Learned Dynamics
Neural Lander: Stable Drone Landing Control using Learned Dynamics
Controller Design (simplified)

• Nonlinear Feedback Linearization:

\[ u_{\text{nominal}} = K_s \eta \]
\[ \eta = \begin{bmatrix} p - p^* \\ v - v^* \end{bmatrix} \]

Feedback Linearization (PD control)

Desired Trajectory (tracking error)

• Cancel out ground effect \( \tilde{F}(s, u_{\text{old}}) \): \[ u = u_{\text{nominal}} + u_{\text{residual}} \]

Requires Lipschitz & small time delay
Controller Design (simplified)

- Nonlinear Feedback Linearization:

\[ u_{\text{nominal}} = K_s \eta \]

\[ \eta = \left[ \begin{array}{c} p - p^* \\ v - v^* \end{array} \right] \]

Desired Trajectory (tracking error)

Stability Guarantee:
(simplified)

\[ ||\eta(t)|| \leq ||\eta(0)|| \exp \left\{ \frac{\lambda_{\text{min}}(K) - \bar{L}\rho}{C} t \right\} + \frac{\epsilon}{\lambda_{\text{min}}(K) - \bar{L}\rho} \]

\[ \Rightarrow ||\eta(t)|| \rightarrow \frac{\epsilon}{\lambda_{\text{min}}(K) - \bar{L}\rho} \quad \text{Exponentially fast} \]
Aside: Robust Regression for Safe Exploration

- Robust regression for provable extrapolation => Safe Exploration!

Robust Regression for Safe Exploration in Control,
Angie Liu, Guanya Shi, Soon-Jo Chung, Anima Anandkumar, Yisong Yue, L4DC 2020

Chance-Constrained Trajectory Optimization for Safe Exploration and Learning of Nonlinear Systems,
Yashwanth Kumar Nakka, Angie Liu, Guanya Shi, Anima Anandkumar, Yisong Yue, Soon-Jo Chung, R-AL 2021

Angie Liu
Yashwanth Nakka
Aside: Learning Control Lyapunov/Barrier Functions

• CLFs & CBFs encode low-dimensional projection of dynamics
  • DOF of action space rather than state space
  • Can be easier to learn than full dynamics

• How to learn CLF/CBF for controller design?
• How to analyze stability/safety under uncertainty?

Episodic Learning with Control Lyapunov Functions for Uncertain Robotic Systems

A Control Lyapunov Perspective on Episodic Learning via Projection to State Stability

Learning for Safety-Critical Control with Control Barrier Functions
Andrew Taylor, Andrew Singletary, Yisong Yue, Aaron Ames. L4DC 2020.

A Control Barrier Perspective on Episodic Learning via Projection-to-State Safety
Summary: Dynamics Learning

• Learn residual dynamics (data efficient)

• Control Lipschitz constant (imposes compatible structure)

• Standard controller design (inherits guarantees)

• Robust regression for safe exploration (provable limited extrapolation)
Integration of Learning at Varying Levels

- Integration in control/action

- Integration in dynamics modeling

- Integration in optimization problem
Model-Based Planning

• Environment model is given

• Design global plan (aka trajectory)

• Satisfy global constraints
  • Previous topics only ensured local constraints
  • E.g., Lyapunov stability, smoothness

• NP-Hard optimization problem!

\[ s_{t+1} = F(s_t, u_t) + \epsilon \]
Optimization as Sequential Decision Making

- Many Solvers are Sequential
  - Tree-Search
  - Greedy
  - Gradient Descent

- Can view solver as “agent” or “policy”
  - State = intermediate solution
  - Find a state with high reward (solution)
  - Learn better local decision making
Learning to Search/Plan

- Learning to Search via Retrospective Imitation [arXiv]
- Co-training for Policy Learning [UAI 2019]

Submodular Maximization

- Learning Policies for Contextual Submodular Prediction [ICML 2013]
- Learning to Make Decisions via Submodular Regularization [ICLR 2021]

Learning to Infer

- Iterative Amortized Inference [ICML 2018]
- A General Method for Amortizing Variational Filtering [NeurIPS 2018]
- Iterative Amortized Policy Optimization [arXiv]
Optimization as Sequential Decision Making

Learning to Search/Plan

- Learning to Search via Retrospective Imitation [arXiv]
- Co-training for Policy Learning [UAI 2019]

Submodular Maximization

- Learning Policies for Contextual Submodular Prediction [ICML 2013]
- Learning to Make Decisions via Submodular Regularization [ICLR 2021]

Learning to Infer

- Iterative Amortized Inference [ICML 2018]
- A General Method for Amortizing Variational Filtering [NeurIPS 2018]
- Iterative Amortized Policy Optimization [arXiv]
While limiting the probability of crashing into obstacles over a cost function.

\[
\min_U \ J(U, X)
\]

subject to,

(Dynamic Constraint) \[ x_{t+1} = Ax_t + Bu_t, \]

(Safety Constraints) \[ h_i^T x_t \leq g_i \]

Compiled as Combinatorial Search Problems

Imitation Learning

Collect Demonstrations

Test Instances

Distribution of Planning Problems

Algorithm 1:

Inputs: \( \mathcal{D} \): expert traces dataset

\( \alpha \): mixing parameter

First, \( \mathcal{D} \) is collected from expert demonstrations and initial policy trained.

Then, the \( \mathcal{D} \) is rolled out to generate its own traces.

Finally, the policy is updated according to the feedback generated.

Algorithm 2:

Inputs: \( s \): search tree trace

\( \sqrt{s} \): parent of \( s \)

Feedback from Retrospective Oracle

Design Decisions.

What constitutes a mistake is also influenced by the actions a policy takes. For example, in (Algorithm 1), a selection \( \mathcal{D} \) is made during each roll-out in order to better imitate the order they are visited.

Algorithm 1 and terminal states, respectively. Numbers in nodes indicate that this example, the search space is organized as a tree where circular and diamond nodes represent intermediate states.
Figure 4: (left) Retrospective imitation versus off-the-shelf methods. The RL baseline performs very poorly due to sparse environmental rewards. (middle, right) Single-step decision error rates, used for empirically validating theoretical claims.

Figure 5: Retrospective DAgger (“select only” policy class) with off-the-shelf branch-and-bound solvers using various search node budgets. Retrospective DAgger consistently outperforms baselines.

Empirically Validating Theoretical Results. Finally, we evaluate how well our theoretical results in Section 5 characterize experimental results. Figure 4b and 4c presents the optimal move error rates for the maze experiment, which validates Proposition 1 that retrospective imitation is guaranteed to result in a policy that has lower error rates than imitation learning. The benefit of having a lower error rate is explained by Theorem 2, which informally states that a lower error rate leads to shorter search time. This result is also verified by Figure 2a and 2d, where Retrospective DAgger/SMILe, having the lowest error rates, explores the fewest number of squares at each problem scale.

7 Conclusion & Future Work

We have presented the retrospective imitation approach for learning combinatorial search policies. Our approach extends conventional imitation learning, by being able to learn good policies without requiring repeated queries to an expert. A key distinguishing feature of our approach is the ability to scale to larger problem instances than contained in the original supervised training set of demonstrations. Our theoretical analysis shows that, under certain assumptions, the retrospective imitation learning scheme is provably more powerful and general than conventional imitation learning. We validated our theoretical results on a maze solving experiment and tested our approach on the problem of risk-aware path planning, where we demonstrated both performance gains over conventional imitation learning and the ability to scale up to large problem instances not tractably solvable by commercial solvers.

By removing the need for repeated expert feedback, retrospective imitation offers the potential for increased applicability over imitation learning in search settings. However, human feedback is still a valuable asset as human computation has been shown to boost performance of certain hard search problems [Le Bras et al., 2014]. It will be interesting to incorporate human computation into the retrospective imitation learning framework so that we can find a balance between manually instructing and autonomously reasoning to learn better search policies. Retrospective imitation lies in a point in the spectrum between imitation learning and reinforcement learning; we are interested in exploring other novel learning frameworks in this spectrum as well.

Our Approach

Gurobi
SCIP
Initial demonstrations only at smallest size!

Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Ongoing: ENav Integration

Shreyansh Daftry
Hiro Ono
Olivier Toupet
Neil Abcouwer
Siddarth Venkatraman
### Preliminary Results

#### Baseline ENav (Cycle Time(s))

<table>
<thead>
<tr>
<th></th>
<th>7%</th>
<th>10%</th>
<th>12%</th>
<th>15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 deg.</td>
<td>1.45</td>
<td>3.38</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>15 deg.</td>
<td>1.00</td>
<td>2.49</td>
<td>2.89</td>
<td>3.58</td>
</tr>
<tr>
<td>10 deg.</td>
<td>0.99</td>
<td>2.39</td>
<td>3.17</td>
<td>2.06</td>
</tr>
<tr>
<td>5 deg.</td>
<td>0.77</td>
<td>2.38</td>
<td>3.79</td>
<td>2.09</td>
</tr>
<tr>
<td>0 deg.</td>
<td>0.98</td>
<td>2.57</td>
<td>3.44</td>
<td>5.18</td>
</tr>
</tbody>
</table>

#### MLNav (Cycle Time(s))

<table>
<thead>
<tr>
<th></th>
<th>7%</th>
<th>10%</th>
<th>12%</th>
<th>15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 deg.</td>
<td>0.80</td>
<td>0.99</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>15 deg.</td>
<td>0.48</td>
<td>0.99</td>
<td>1.20</td>
<td>1.45</td>
</tr>
<tr>
<td>10 deg.</td>
<td>0.47</td>
<td>1.20</td>
<td>1.38</td>
<td>1.41</td>
</tr>
<tr>
<td>5 deg.</td>
<td>0.54</td>
<td>0.93</td>
<td>1.25</td>
<td>1.28</td>
</tr>
<tr>
<td>0 deg.</td>
<td>0.47</td>
<td>1.11</td>
<td>1.33</td>
<td>1.78</td>
</tr>
</tbody>
</table>

---

Learned Decentralized Planner (enforcing safety)

GLAS: Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning with End-to-End Learning, Benjamin Rivière, et al., R-AL 2020
5. Deploy: Six robots navigating an obstacle course.
Blending Models/Rules & Black-Box Learning

Further Research!
Collaborators

Smooth Imitation Learning for Online Sequence Prediction, Hoang Le, et al., ICML 2016
Control Regularization for Reduced Variance Reinforcement Learning, Richard Cheng et al. ICML 2019
Batch Policy Learning under Constraints, Hoang Le, et al. ICML 2019
Imitation-Projected Programmatic Reinforcement Learning, Abhinav Verma, Hoang Le, et al., NeurIPS 2019
Neural Lander: Stable Drone Landing Control using Learned Dynamics, Guanya Shi, et al., ICRA 2019
Neural-Swarm: Decentralized Close-Proximity Multirotor Control Using Learned Interactions, Guanya Shi et al., ICRA 2020
Neural-Swarm2: Planning and Control of Heterogeneous Multirotor Swarms using Learned Interactions, Guanya Shi et al., arXiv
Robust Regression for Safe Exploration in Control, Angie Liu, Guanya Shi, et al., L4DC 2020
Episodic Learning with Control Lyapunov Functions for Uncertain Robotic Systems, Andrew Taylor, Victor Dorobantu, et al., IROS 2019
A Control Lyapunov Perspective on Episodic Learning via Projection to State Stability, Andrew Taylor, Victor Dorobantu, et al., CDC 2019
Learning for Safety-Critical Control with Control Barrier Functions, Andrew Taylor, et al., L4DC 2020
A Control Barrier Perspective on Episodic Learning via Projection-to-State Safety, Andrew Taylor, et al., L-CSS 2020
Learning to Search via Retrospective Imitation, Jialin Song, Ravi Lanka, et al., arXiv
Co-Training for Policy Learning, Jialin Song, Ravi Lanka, et al., UAI 2019
GLAS: Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning with End-to-End Learning, Benjamin Rivière, et al., R-AL 2020
Learning Policies for Contextual Submodular Optimization, Stephane Ross et al., ICML 2013
Learning to Make Decisions via Submodular Regularization, Ayya Alieva, Aiden Aceves, et al., ICLR 2021
Iterative Amortized Inference, Joe Marino et al., ICML 2018
A General Framework for Amortizing Variational Filtering, Joe Marino et al, NeurIPS 2018
Iterative Amortized Policy Optimization, Joe Marino et al., arXiv
Machine Learning Based Path Planning for Improved Rover Navigation, Neil Abcouwer et al., IEEE AeroConf 2021