Cautious Adaptation For Reinforcement Learning in Safety-Critical Settings

Zhang et al. (2020)

presented by Maren Eberle

Advanced Topics in Reinforcement Learning

08.12.2022

Outline

Figure 1: Safety-Critical Adaptation Framework (adapted: Zhang et al. 2020, p. 1).

- Intuition: Human behavior
- Goal: Safe during learning in target environment

☞ Act with caution

Zhang et al. (2020)

Framework

Idea
Safety-Critical Adaptation (SCA)

1. Pretraining: Prior experience with uncertainty and risk from sandbox environments

☞ knowledge transfer

2. Adaptation: Adapt to a safety-critical target environment

Approach
Cautious Adaptation in RL (CARL)

1. Pretraining: Train model on action score in unknown environments

☞ risk-averse exploration

2. Adaptation: Train model on modified action score in unknown environment

☞ Safe RL reduced to safe adaptation

Zhang et al. (2020)

Placement in Safe RL

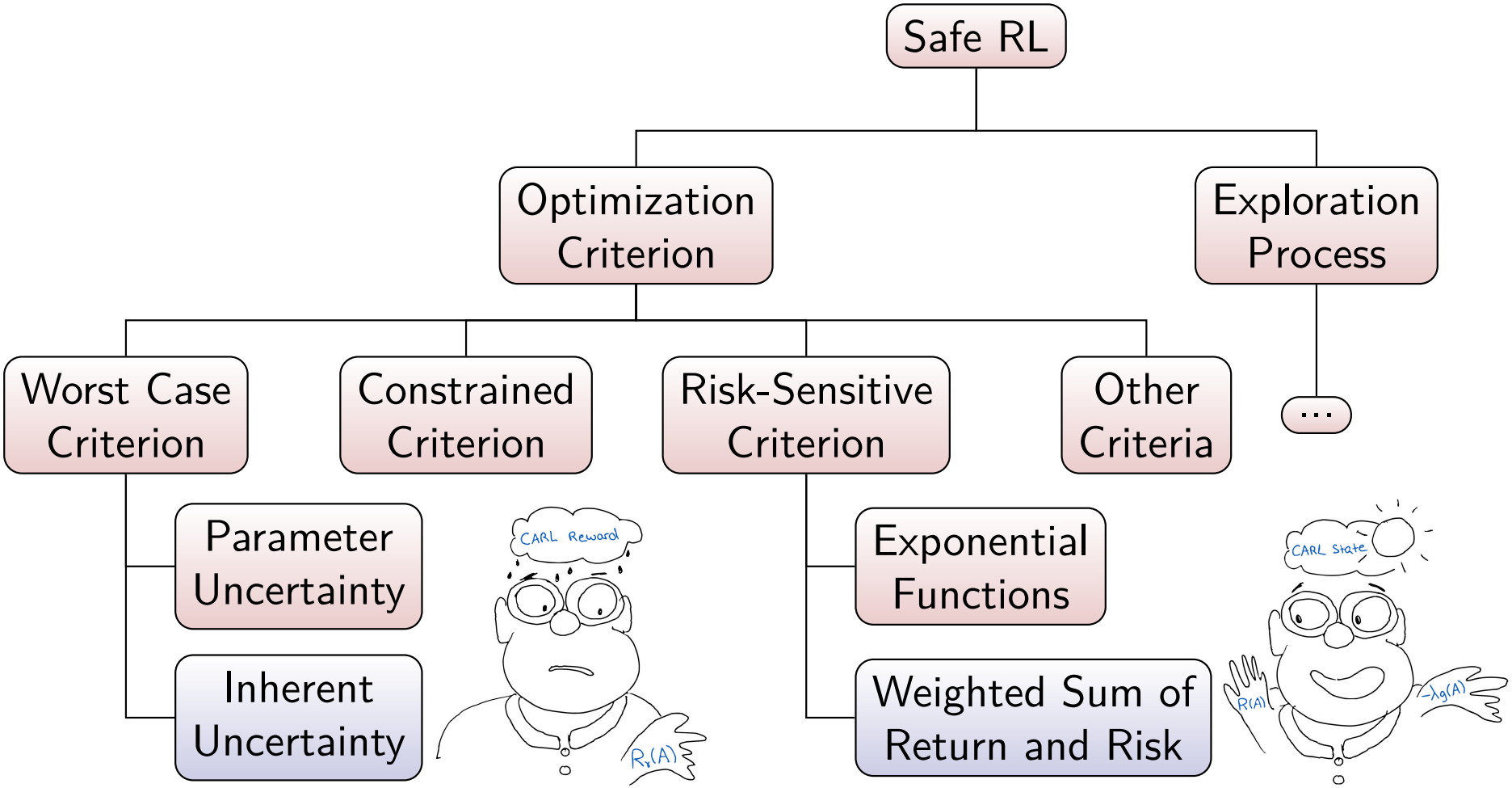


Figure 2: Approaches for Safe RL (adapted: García and Fernández 2015, p. 1440).

Background: Probabilistic Ensembles with Trajectory Sampling (PETS)

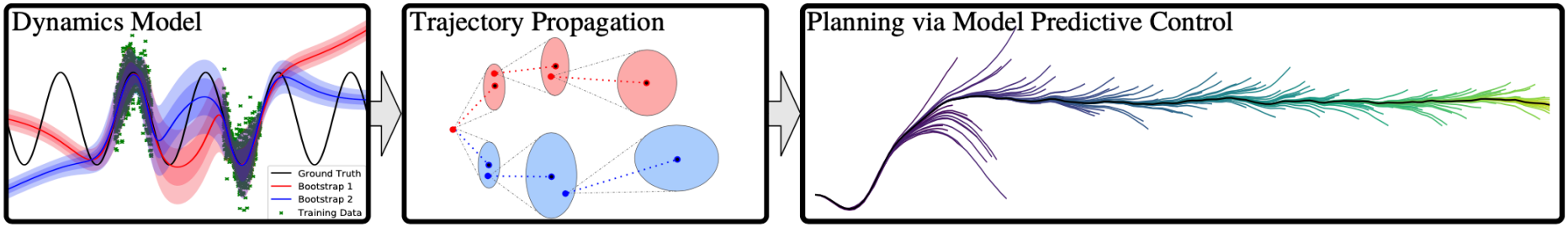


Figure 3: PETS method (Chua et al. 2018, p. 2).

	Epistemic Uncertainty	Aleatoric Uncertainty
Origin	limited data	system stochasticity
Solution	ensemble (bootstrapping)	probabilistic neural network

Table 1: Model uncertainties captured by Dynamics Model.

Chua et al. (2018)

Terminology

- $A = [a_1, ..., a_H]$ action sequence
  - H planning horizon
- $s_0, ...$ states

- f dynamics model

☞ $\{\hat{s}_H^i\}_{i=1}^N$ particles (samples from distribution of possible states)
  - N number of particles $i \in [1, N]$
  - r^i reward for particle i
  - $s_i = f(s_{i1}, a_{i1})$

- g state safety model

☞ $w_1, ...$ state safety label

Zhang et al. (2020)

Modeling Uncertainties

- Model-based
- Action score from PETS
- One PETS agent trained across several environments
- Catastrophic actions allowed

(1) Action score

$$R(A) = \sum_{i=0}^{N} r^i / N$$

Zhang et al. (2020)

Pretraining Algorithm

1. initialize f, g and train them on random trial

☞ at step t

2. sample action sequence A until planning horizon H
3. produce N particles for A
4. calculate $R(A)$ (1)

☞ repeat for other A

- find $A^* = argmax(R(A))$
- take first action in A^*
- add $(s_t, a_t, s_{t+1}, w_{t+1})$ to data

☞ repeat for every step until task horizon

5. train f, g on data

☞ repeat in every training environment

Zhang et al. (2020)

Low Reward Risk-aversion

- CARL Reward
- Pessimistic evaluation of action sequences
- Notion of risk: Producing low rewards

☞ "low reward ≈ catastrophic action"

(2) Generalized Action Score

$$R_\gamma(A) = \sum_{i:r^i \leq v_{100-\gamma}(r)} r^i / N$$

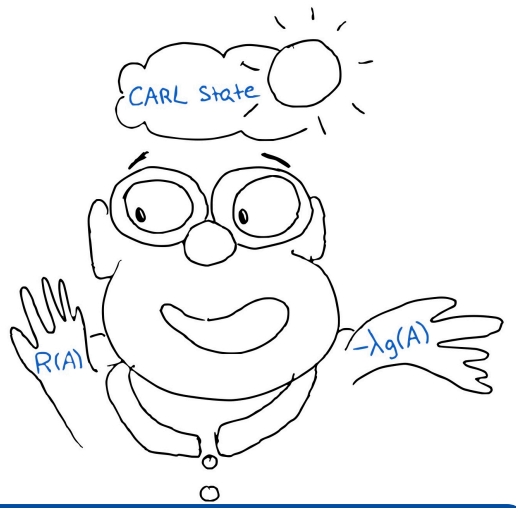
with caution parameter γ and $v_k(r)$ value of k^{th} percentile of rewards

Zhang et al. (2020)

Catastrophic State Risk-aversion

- CARL State
- Predict and penalize probability of encountering catastrophe
- Notion of risk: State safety (probability of encountering a catastrophic state)

☞ "reward $\neq$ state safety"



(3) Action Score with Penalty

$$R_\lambda(A) = R(A) - \lambda g(A)$$

with penalty weight λ and predicted catastrophe cost $g(A)$

Zhang et al. (2020)

Adaptation Algorithm

1. use f, g and data from pretraining

☞ at step t

2. sample action sequence A until planning horizon H
3. produce N particles for A
4. calculate $R_\gamma(A)$ (2) or $R_\lambda(A)$ (3)

☞ repeat for other A

- find $A^* = argmax((2) \text{ or } (3))$
- take first action in A^*
- add $(s_t, a_t, s_{t+1}, w_{t+1})$ to data

5. finetune f on data

☞ repeat for every step until task horizon

Zhang et al. (2020)

Evaluation Criteria for Adaptation

Goal	Analysis Method
capture epistemic uncertainties	plot trajectories during pretraining
high task rewards	average maximum reward
minimal catastrophic events	total number of catastrophes
quick adaptation	plot results over adaptation time

Table 2: Evaluation criteria for CARL only / SCA algorithms in general.

- Expectation: Cautious behavior is most pertinent when the target environment is significantly different

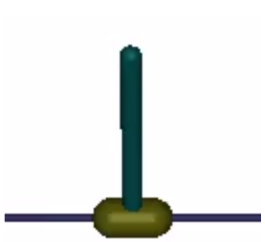
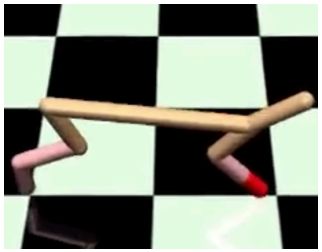
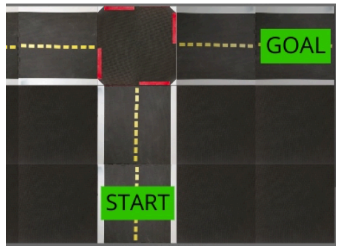
Zhang et al. (2020)

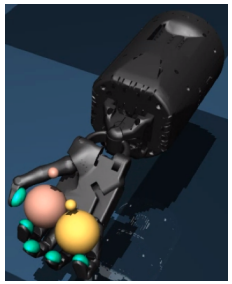
SCA Baseline Algorithms

- PPO-MAML: MAML (Model-Agnostic Meta-Learning) applied to PPO-trained agent
  - Model-free
  - Metalearning during pretraining
  - 1500x CARL pretraining episodes
- RARL: Robust Adversarial Reinforcement Learning
  - Model-free
  - 2x CARL pretraining episodes
  - 20x CARL pretraining episodes
- MB + Finetune: PETS for pretraining and adaptation
  - Model-based
  - Same pretraining as CARL

Zhang et al. (2020)

Target Environments

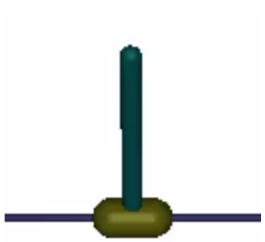
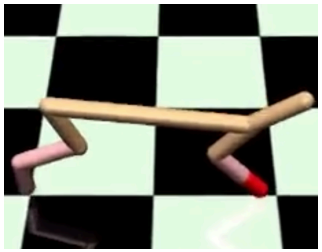
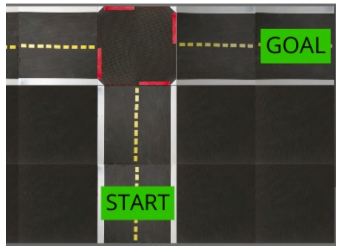


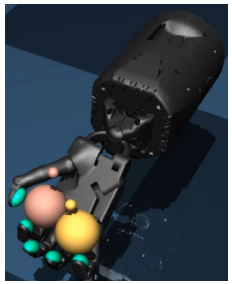
	CartPole	Half-cheetah	Car-driving	Hand manipulation
pretraining model	PETS	PETS	PETS	PDDM
adaptation episodes	10	10	10	15
rollouts per episode	1	1	1	35

Table 3: Comparison of target environments.

Zhang et al. (2020)

Target Environments



	CartPole	Half-cheetah	Car-driving	Hand manipulation
pretraining model	PETS	PETS	PETS	PDDM
adaptation episodes	10	10	10	15
rollouts per episode	1	1	1	35
PPO-MAML	✓	✓	✗	✗
RARL 2x	✓	✓	✓	✗
RARL 20x	✓	✓	✓	✗
MB + Finetune	✓	✓	✓	✓

Table 3: Comparison of target environments.

Zhang et al. (2020)

Rewards and Catastrophes

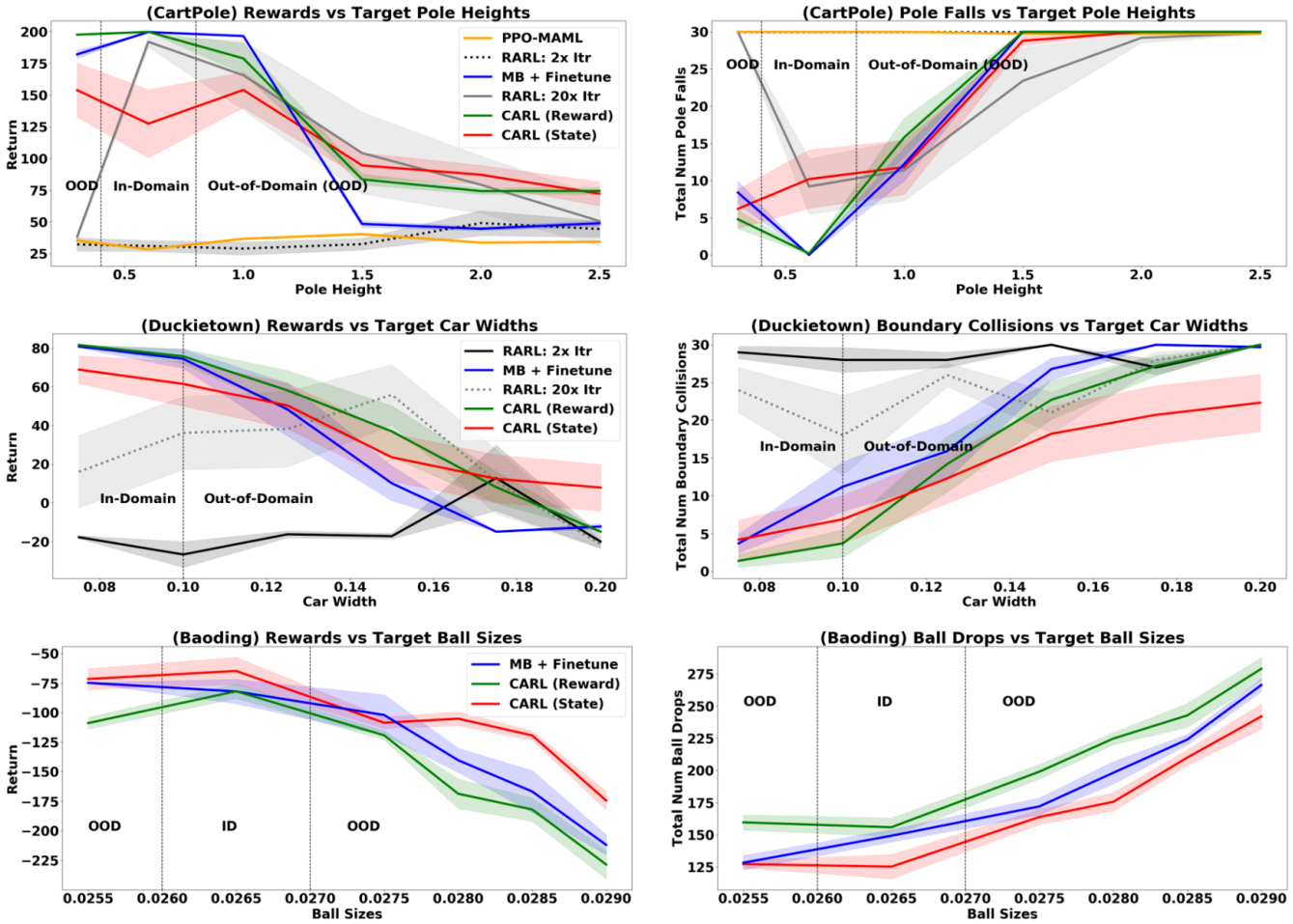


Figure 4: Average maximum reward and number of catastrophic events over domain similarity (Zhang et al. 2020, p. 8).

Adaptation Speed

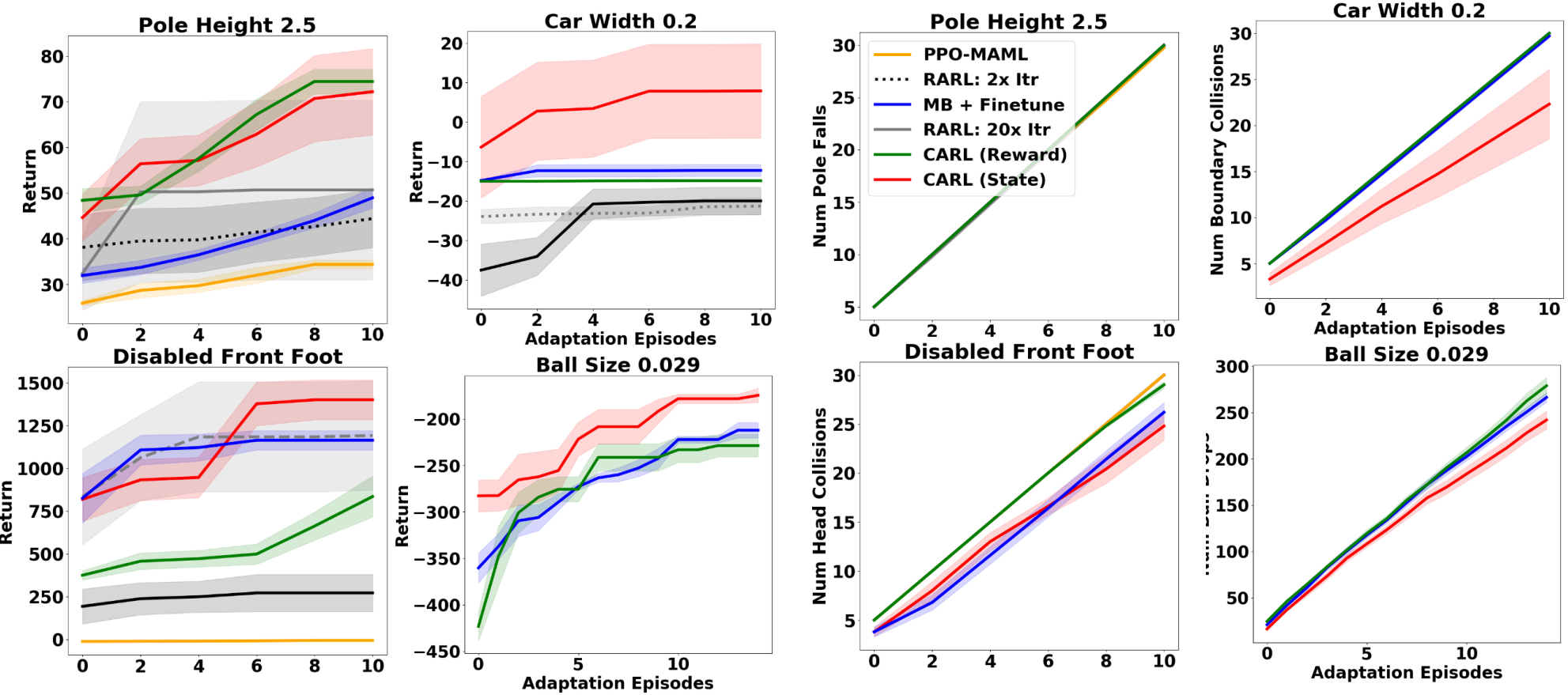
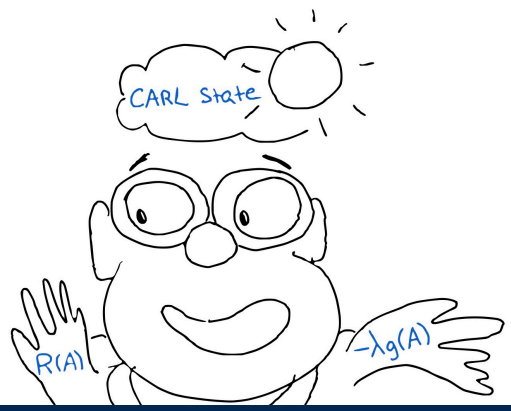


Figure 5: Average maximum reward and number of catastrophic events over time (adapted: Zhang et al. 2020, p. 9).

Summary

- CARL State usually performs better than CARL Reward
- ☞ "reward $\neq$ state safety" is a more useful notion of risk
- Compared to baselines, CARL achieves high rewards, few catastrophic events and fast learning combined
- CARL can be used on other pretraining algorithms than PETS, too

Zhang et al. (2020)



Outlook

- Speed-safety tradeoff

inspired by https://openreview.net/forum?id=BkxA5lBFvH

- Hyperparameter tuning (ensemble size, hidden layers, planning horizon,...) (Zhang et al. 2020)
- ☞ Overall, CARL State is a good example algorithm for the SCA framework
- Specifics of finetuning during adaptation
- Difficulty/similarity of sandbox vs target environments not adressed
  - Here, sandbox and target environment are almost the same (e.g. different car/ ball size than target)
  - How would this scale with complexity?

Questions?

References

Chua, Kurtland et al. (2018). *Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models*. DOI: 10.48550/ARXIV.1805.12114. URL: https://arxiv.org/abs/1805.12114.

García, Javier and Fernando Fernández (2015). "A Comprehensive Survey on Safe Reinforcement Learning". In: *J. Mach. Learn. Res.* 16.1, pp. 14371480. ISSN: 1532-4435.

Zhang, Jesse et al. (2020). "Cautious Adaptation For Reinforcement Learning in Safety-Critical Settings". In: DOI: 10.48550/ARXIV.2008.06622. URL: https://arxiv.org/abs/2008.06622.

Catastrophe Probability Prediction

(4) Predicted Catastrophe Cost

$$g(A) = \sum_{i=1}^{H} P(s_i \in CatastrophicSet)$$

(5) Probability of Catastrophe

$$P(s_i \in CatastrophicSet) = \frac{\sum_{j=1}^{|E|} \delta(c_{\theta_j}(s_{i-1}, a_{i-1}) > \beta)}{|E|}$$

with $|E|$ number of environments, $\{\theta_1, ..., \theta_{|E|}\}$ ensemble parameters, caution tuning parameter $\beta \in \{0.25, 0.5, 0.75\}$

- *CatastrophicSet* has to be defined for each environment

Zhang et al. (2020)

Pretraining Algorithm

Algorithm 1 Pretraining

Initialize probabilistic ensemble dynamics model f and state safety model g.
Collect data $\mathcal{D}$ by executing a random controller in one random training environment for one trial.
for environment ID $z \sim$ training environments **do**
    Train the models f and g to predict state transitions and state safety respectively, on $\mathcal{D}$
    for $t = 0$ to task horizon **do**
        for evolutionary search stage=1,2,... **do**
            for sampled action sequence A **do**
                Run state propagation to produce N particles
                Evaluate A as $R(A) = \sum_i r_i / N$
            end for
            Refine search to find $A^* = \arg\max R(A)$
        end for
        Set a_t to first action of A^*, and execute a_t.
        Record the state transition and state safety label (w_{t+1}) as a tuple $(s_t, a_t, s_{t+1}, w_{t+1})$ in $\mathcal{D}$
    end for
end for

Figure 6: CARL pretraining algorithm (Zhang et al. 2020, p. 4).

Adaptation Algorithm

Algorithm 2 Adaptation

Inputs: Pretraining dataset $\mathcal{D}$
for target environment adaptation episode=1,2,... **do**
  for $t = 0$ to task horizon **do**
    for evolutionary search stage=1,2,... **do**
      for sampled action sequence A **do**
        Run state propagation
        Evaluate A with generalized score (Eq 2 or 3)
      end for
      Refine search to find $A^* = \arg\max R_\gamma(A)$
    end for
    Execute first action of A^*
    Record outcome in $\mathcal{D}$
    Finetune the probabilistic ensemble model f on $\mathcal{D}$
  end for
end for

Figure 7: CARL adaptation algorithm (Zhang et al. 2020, p. 5).

Pretraining: Capturing Uncertainty

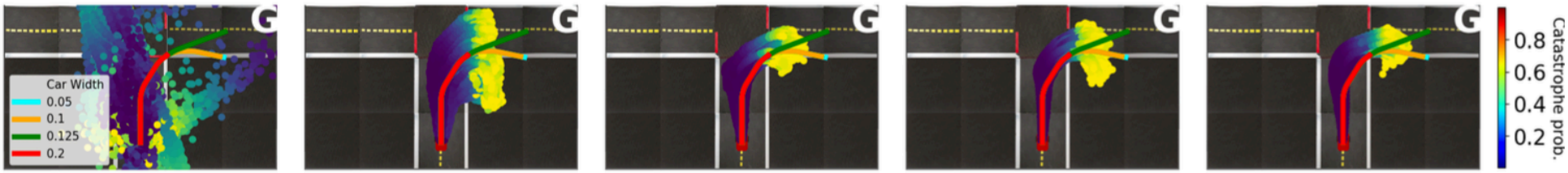


Figure 8: Capturing epistemic uncertainty during pretraining with CARL (Zhang et al. 2020, p. 7).

Planning with Deep Dynamics Models (PDDM)

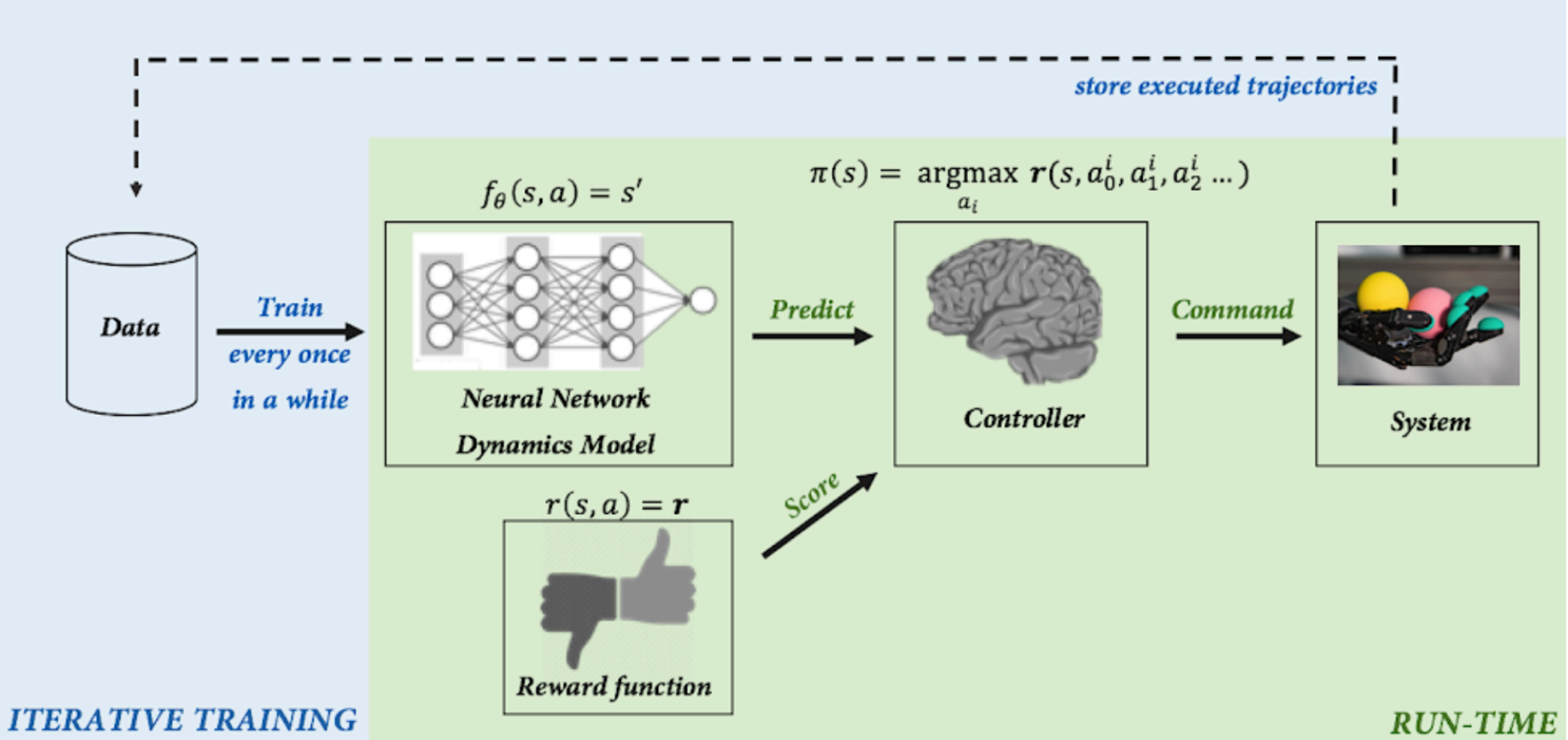


Figure 9: Overview of PDDM algorithm for online planning with deep dynamics models (Nagabandi 2019, https://bair.berkeley.edu/blog/2019/09/30/deep-dynamics/).