A. CARL Catastrophe Probability Prediction

Constraints and Formulation Our planner starts from a state s_0, a sampled action sequence $A = [a_1, ..., a_H]$ over a planning horizon H, our ensemble parameters $\{\theta_1, ..., \theta_{|E|}\}$, a caution tuning parameter β, a function f parameterized by θ that maps states and actions to distributions over the next state, and a function c parameterized by θ that produces collision probabilities.

We denote the predicted catastrophe cost as $g(A)$, which is the predicted sum of failure probabilities for all states along the trajectory produced by A. Similarly to prior work, we define failure probabilities as the probability that the state is in a set of catastrophic states, $CatastrophicSet$, for the given environment. Note that because f outputs distributions over the next state, the states s_i themselves are random variables.

$g(A)$ can then be decomposed into

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

where

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

and

$$s_i = f(s_{i-1}, a_{i-1})$$

That is, the probability of catastrophe is given by the proportion of models in the ensemble that predict a catastrophe to occur with probability greater than β. Here, we are assuming that the true probability of catastrophe can be approximated by the empirical bootstrap distribution produced by our dynamics models. The models output isotropic Gaussian distributions over the next state and the probability of catastrophe given the previous (s_{i-1}, a_{i-1}) pair. In practice, we do not need to divide by the ensemble size $|E|$, as the λ coefficient on $g(A)$ in Eq. 3 can be scaled appropriately to absorb the coefficient.

Connection to Constrained MDPs We can formulate the problem as a chance constrained MDP, where we set a bound on the probability that one or more states is in the catastrophic set. This corresponds to the constraint $\sum_i P(s_i \in CatastrophicSet) <= K$ for all planned states i with a large constant K. The Lagrangian relaxation of this is the exact problem that we optimize for in MPC planning: maximizing Eq. 3.

Training For all of our experiments, we heuristically select β to be 0.5. Furthermore, we set λ used to scale $g(A)$ in $R_\lambda(A)$ to be an arbitrarily high penalty value: $10000 \cdot |E|$. Empirically, in order to train c, we simply extend the state space of our environments with a catastrophe indicator and train the catastrophe output of our dynamics models with the binary cross entropy loss function. Therefore our loss function is the addition of the cross entropy loss term with standard supervised probabilistic state loss term from PETS (Chua et al., 2018).

Ablation on β We perform an ablation on β in appendix Fig B. We see that across CartPole, Duckietown, and Half-Cheetah, setting $\beta = 0.5$ produces the best overall results. Interestingly, setting β lower doesn't necessarily result in fewer catastrophes. All three levels of β perform similarly in the simpler CartPole environment and the challenging Baoding environment.

However in Duckietown we see that $\beta = 0.25$ performs significantly worse than the other two. We notice that this behavior comes from the agent often refusing to make the turn at all, being overly risk-averse about making any action that could lead to hitting the right road tile boundaries. This occasionally leads to the agent then hitting other road boundaries that were not encountered during training time. And as expected, $\beta = 0.75$ experiences the largest number of boundary collisions in out of domain settings as the car width increases. Note that for both CartPole and Duckietown, $\beta = 0.75$ performs the best in-domain, which is to be expected as there is less need to be risk-averse in environments it was trained on.

In Half-Cheetah, $\beta = 0.25$ does allow the agent to start with a higher reward than the other two β ablations, despite encountering a similar number of head collisions. This means the most risk-averse ablation prevents head collisions for longer than the other two methods in the first few stages of adaptation. However the best final performance is achieved by $\beta = 0.5, 0.75$.

In Baoding, performance is generally similar across all three β values.

B. CARL Experiment Details

All model-based methods (CARL `CARL (Reward)`, `MB + Finetune`)) are trained with the same hyperparameters, which are listed in Table 1. For CartPole and Half-Cheetah, the params are chosen with little modification to the original training parameters of the similar environments in PETS (Chua et al., 2018). And for Baoding, the parameters were chosen to reproduce the PDDM (Nagabandi et al., 2019) pretraining results as closely as possible with fixed ball weights, and are based on the ones listed in the paper. For all environments, the number of training iterations is relatively high to ensure the method is able to successfully solve the task and ensure CARL has sufficient training iterations to accurately predict catastrophe probabilities. We find that training for longer produces even better pretraining and adaptation rewards for the Baoding task, however the wall-clock time for both pretraining and adaptation greatly increases. The hyperparameters have not been extensively tuned and it is likely that the parameters and loss function weighting can be adjusted to achieve better adaptation performance with fewer training iterations. We note that the model free methods we compare against are trained on many more iterations.

We train `RARL` on both 2x and 20x the number of iterations we train the model-based methods on for a fair comparison, and we train `PPO-MAML` on 1500x the number of iterations to ensure good meta-training performance before testing adaptation.

Param/Env	CartPole	Half-Cheetah	Duckietown	Baoding
Ensemble Size	5	5	5	3
# Hidden Layers	4	5	4	2
Hidden Layer Size	500	200	200	500
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	1e-3	1e-3	1e-3	1e-3
Batch Size	256	256	256	512
# Random Itrs	1	1	2	100 initial, then 5 per itr
# On-Policy Rollouts	50	100	100	3000 (30 per itr, 100 itrs)
Epochs per Itr	5	10	5	40
Planning Horizon	25	10	25	8
CEM Popsize	400	500	400	-
CEM # Elites	40	50	40	-
β (CARL and CARL (Reward))	0.5	0.5	0.5	0.5

Table 1. Table of parameters for every environment across all model-based methods. See (Nagabandi et al., 2019) or our code for Baoding optimizer parameters.

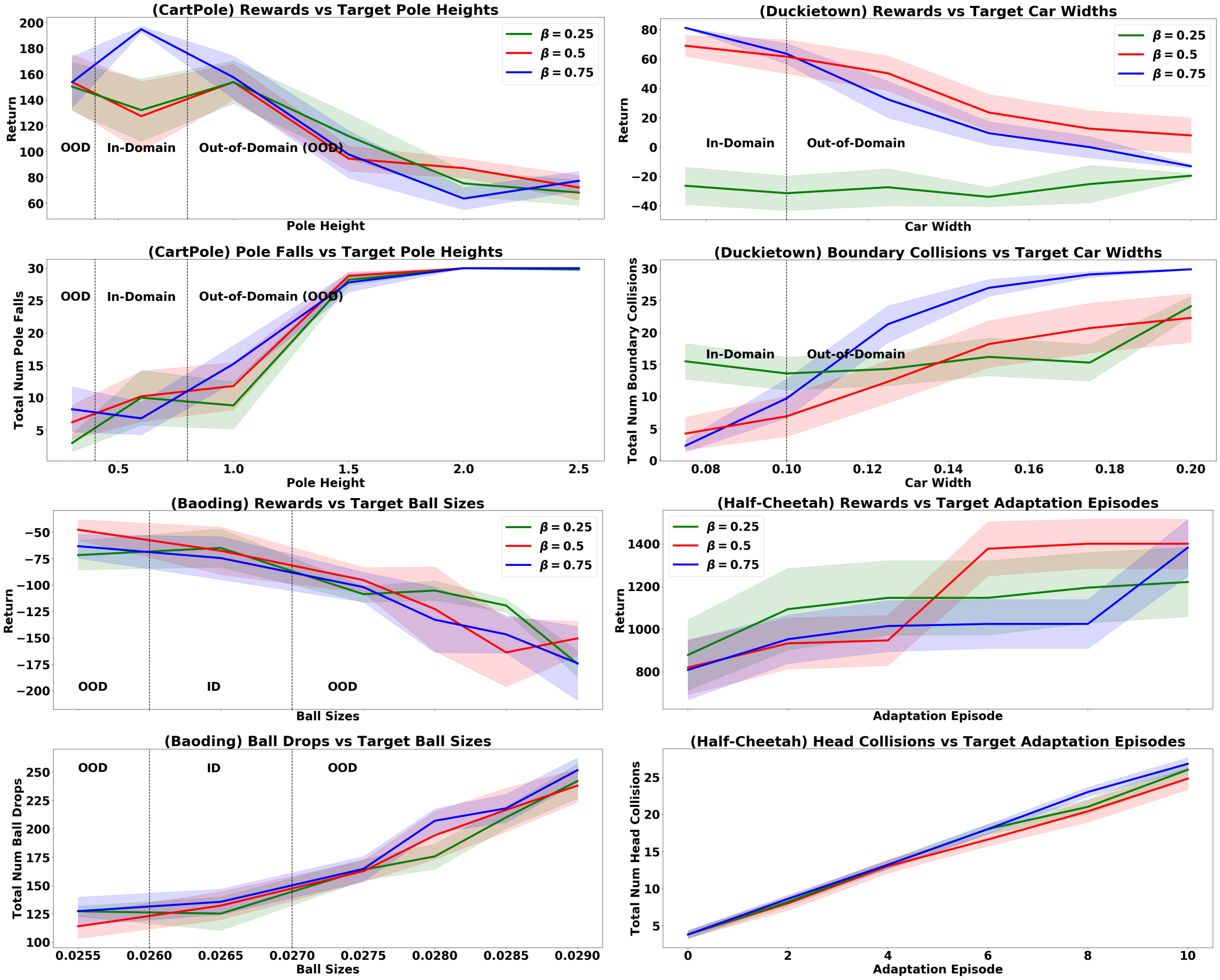

Figure 7. Evaluation of how β affects CARL performance in all 4 environments. A higher β indicates less cautiousness, as the threshold for positively predicting catastrophe by each individual model in the ensemble increases, and vice versa.

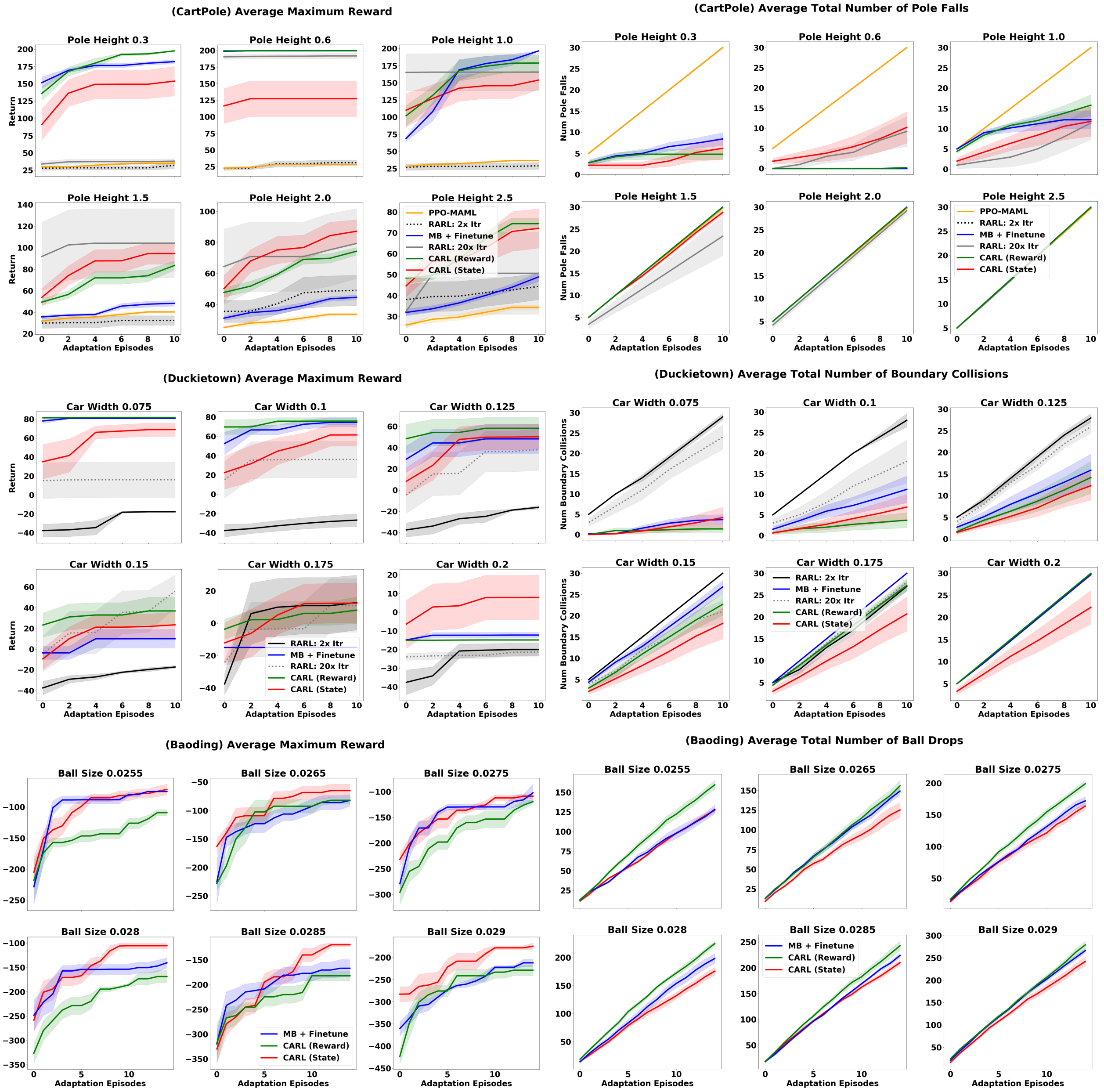

Figure 8. Evaluation of the total number of catastrophes and average rewards over time at all adaptation domains in CartPole, Duckietown, and Baoding. The maximum number of catastrophes is 30 in CartPole and Duckietown, as there are 6 evaluation steps (1 before adaptation starts, 5 during) and at each evaluation step 5 evaluations are performed. The maximum catastrophe number in Baoding is 450, as there are 15 evaluation steps (1 at each adaptation iteration), with 30 evaluations performed. This cumulative total is averaged over ten initializations for each model. Each reward is calculated by taking the maximum reward seen so far at each timestep, where the value at that timestep is averaged over the evaluations. This maximum is then averaged over ten initializations for each model. Standard errors are shown with colored bars.