Uncertainty-Aware Reinforcement Learning for Collision Avoidance

Gregory Kahn et al.

SeungWoon Kim

KAIST

Real-Time 3D Navigation for Autonomous Vision-Guided MAVs / Seungwon Song

- Simplify Quadrotor dynamic
- Reduce resolution of Octomap (octants)

- Octree-Based State Lattice
  - Adjacency between octree node states
  - Multi-resolution path lookup-table
  - Pre-discretization
- Local 3D State Lattice
- Graph search
  - Optimal path finding
  - Path reconstruction

Contents

- Motivation
- Background
- Main Idea
- Results
- Discussion
- Summary and Q&A

Motivation (1)

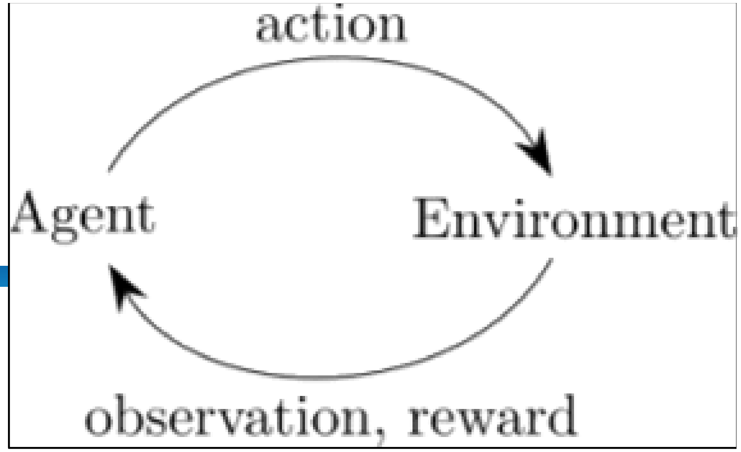
- □ **Policy search via Reinforcement Learning is used in many robotic tasks**
  - ○ **Self-driving vehicles**
  - ○ **Drones**

http://iranjavan.net/wp-content/uploads/2016/08/wdd2.jpg

http://geekongadgets.com/wp-content/uploads/2016/09/Drone.jpg

KAIST

Motivation (2)



□ Reinforcement Learning

- Many RL : Experience failures at training time
- Other RL : Ensure safety by assuming complete state and environment knowledge at training time

  → can restrict the feasibility of real-world robot deployment

In safety-critical domains, choosing proper RL method is important.

Motivation (3)

- ☐ How robot like quadrotor and RC-car avoid obstacles without collision?
- ☐ To avoid obstacle, the robot trains itself by experiencing collision

Problem statement

- **How to do reinforcement learning without destroying the robot during training using only images?**

- **Uncertainty-aware collision prediction model**
  - **Enable a robot to learn how to accomplish a task in unknown environment**
  - **While only experiencing gentle collisions**

KAIST

Background

- Model-free method
  - Simplicity and favorable computational property
- Model-based method
  - Sample-efficient

This approach adopt a model-based learning, learn uncertainty-aware collision avoidance model

Contribution (1)

Risk-averse collision prediction probability

$$\widetilde{P}_\theta(\mathrm{COLL}|\mathbf{x}_t, \mathbf{u}_{t:t+H}, \mathbf{o}_t) =$$

Variance Value (model is certain)

$$\mathrm{L}\left(\mathbb{E}[f_\theta(\mathbf{x}_t, \mathbf{u}_{t:t+H}, \mathbf{o}_t)] + \lambda_{\mathrm{STD}}\sqrt{\mathrm{Var}[f_\theta(\mathbf{x}_t, \mathbf{u}_{t:t+H}, \mathbf{o}_t)]}\right)$$

Expected Value (predict collision) Non-negative user-defined scalar

Cost of Task

$$\mathcal{C}(\mathbf{x}_{t+H}, \mathbf{u}_{t+H}) \approx \mathcal{C}_{\mathrm{TASK}}(\mathbf{x}_{t+H}, \mathbf{u}_{t+H}) + \widetilde{P}_\theta(\mathrm{COLL}|\mathbf{x}_t, \mathbf{u}_{t:t+H}, \mathbf{o}_t)\mathcal{C}_{\mathrm{COLL}}(\mathbf{x}_{t+H})$$

Collision probability function

- x_t : Current state / u_t : Action / o_t : Observation

KAIST

Contribution (2)

- ☐ **Uncertainty aware model-based RL**
  - ○ **Uses bootstrapping and dropout to yield actionable uncertainty estimates**
  - ○ **Process raw sensory inputs such as camera etc.**

- ☐ **Why dropout and bootstrapping?**
  - ○ **Dropout : can estimate uncertainty for regression tasks such as motor control**
  - ○ **Bootstrapping : likely to estimate high uncertainty in novel environments**

KAIST

Contribution (2)

Algorithm 1 Neural net training with bootstrapping and dropout

1: **input**: dataset $\mathcal{D} = \{\mathbf{x}_t^{(i)}, \mathbf{u}_{t:t+H}^{(i)}, \mathbf{o}_t^{(i)}\}$, neural network model NN
2: **for** $b = 1$ to B **do**
3: Sample a dataset of subsequences $\mathcal{D}^{(b)}$ from the full dataset $\mathcal{D}$ with replacement
4: Initialize neural network $\text{NN}^{(b)}$ with random weights
5: **for** number of SGD iterations **do**
6: Sample datapoint $(\mathbf{x}_t, \mathbf{u}_{t:t+H}, \mathbf{o}_t)$ from $\mathcal{D}^{(b)}$
7: Sample $\text{NN}_d^{(b)}$ by masking the units in $\text{NN}^{(b)}$ using dropout
8: Run forward pass on $\text{NN}_d^{(b)}$ using $(\mathbf{x}_t, \mathbf{u}_{t:t+H}, \mathbf{o}_t)$
9: Run backward pass on $\text{NN}_d^{(b)}$ to get gradient $g_d^{(b)}$
10: Update model $\text{NN}^{(b)}$ parameters using $g_d^{(b)}$
11: **end for**
12: **end for**

Bootstrapping (3)

Dropout (7)

Gradient updates (8~10)

Main Idea (1)

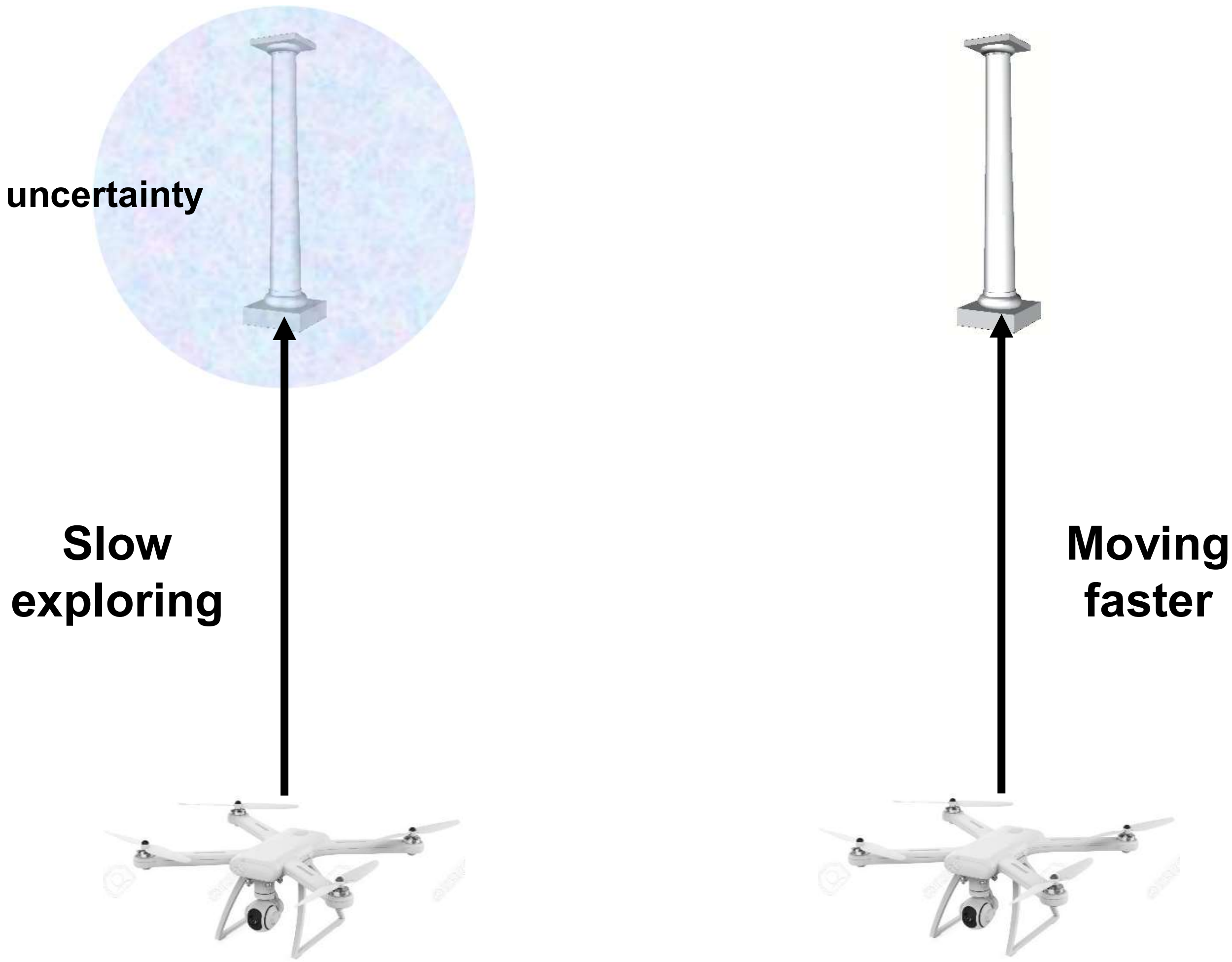
□ Approach

- Uncertainty-aware collision prediction model, Speed-dependent collision cost
- When uncertainty is high → exploring cautiously
  When uncertainty is low → moving faster
- Input : image, a sequence of velocity commands
- Output : the probability of collision
- Goal : avoid obstacles in an unknown environment

Algorithm 2 RL with Risk-Averse Collision Estimates

Initialize empty dataset $\mathcal{D}$
Initialize collision prediction model $\widetilde{P}_\theta$
for iter=1 to max_iter **do**
  Sample trajectories $\{\tau_i\}$ using MPC with cost C
  Add samples $\{\tau_i\}$ to $\mathcal{D}$
  Train $\widetilde{P}_\theta$ using $\mathcal{D}$ (Alg. 1)
end for

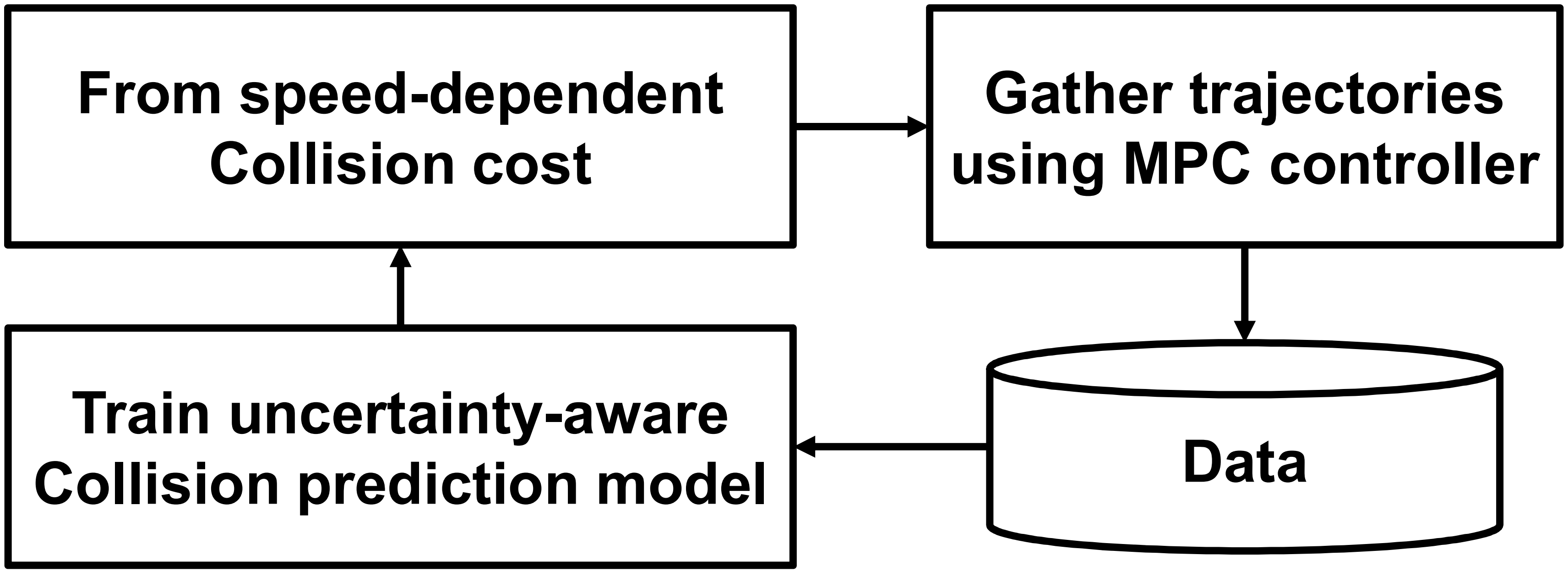
Main Idea (2)



Main Idea (3)

□ Model-based RL Algorithm

Experience safe, low-speed collisions by reasoning about the model's uncertainty



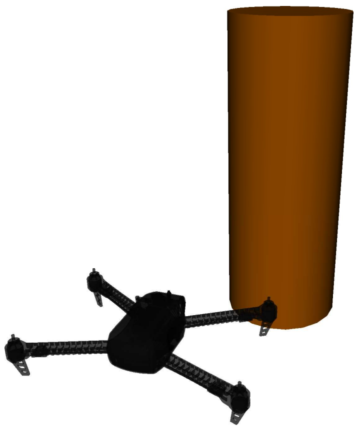
Deep neural network with uncertainty estimates from bootstrapping and dropout

Robot increases speed as model becomes more confident

Results (1)

☐ Experiments

- ○ Task : Navigating in an unknown environment without collision
- ○ Object : Quadrotor, RC-car
- ○ Environments : Simulated and Real-world

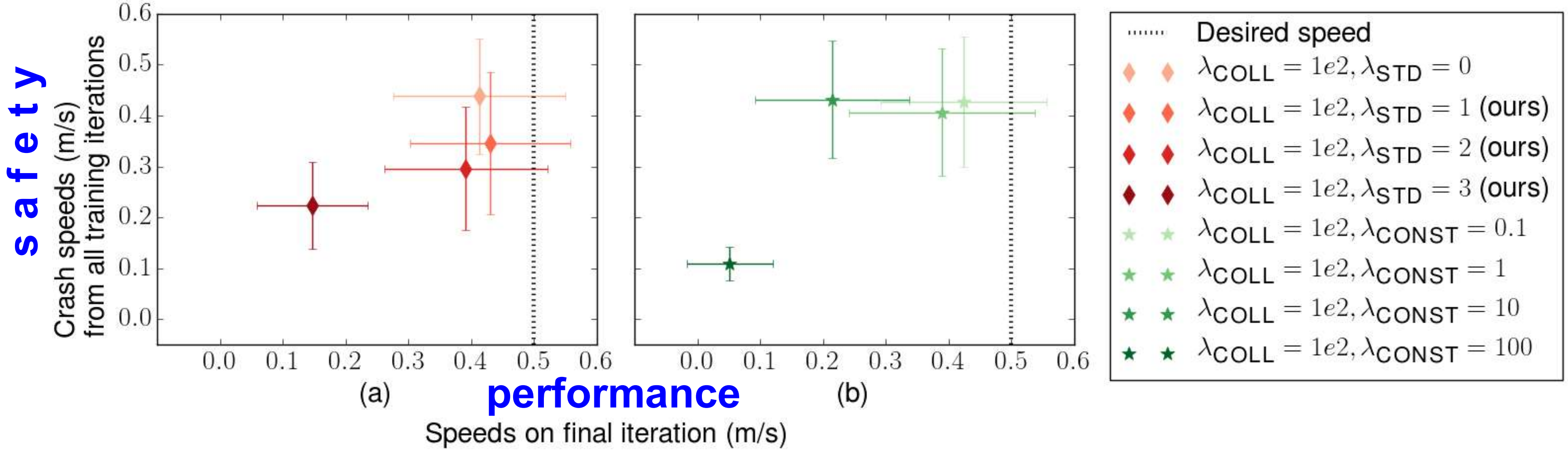


KAIST

Results (2)

□ Quadrotor

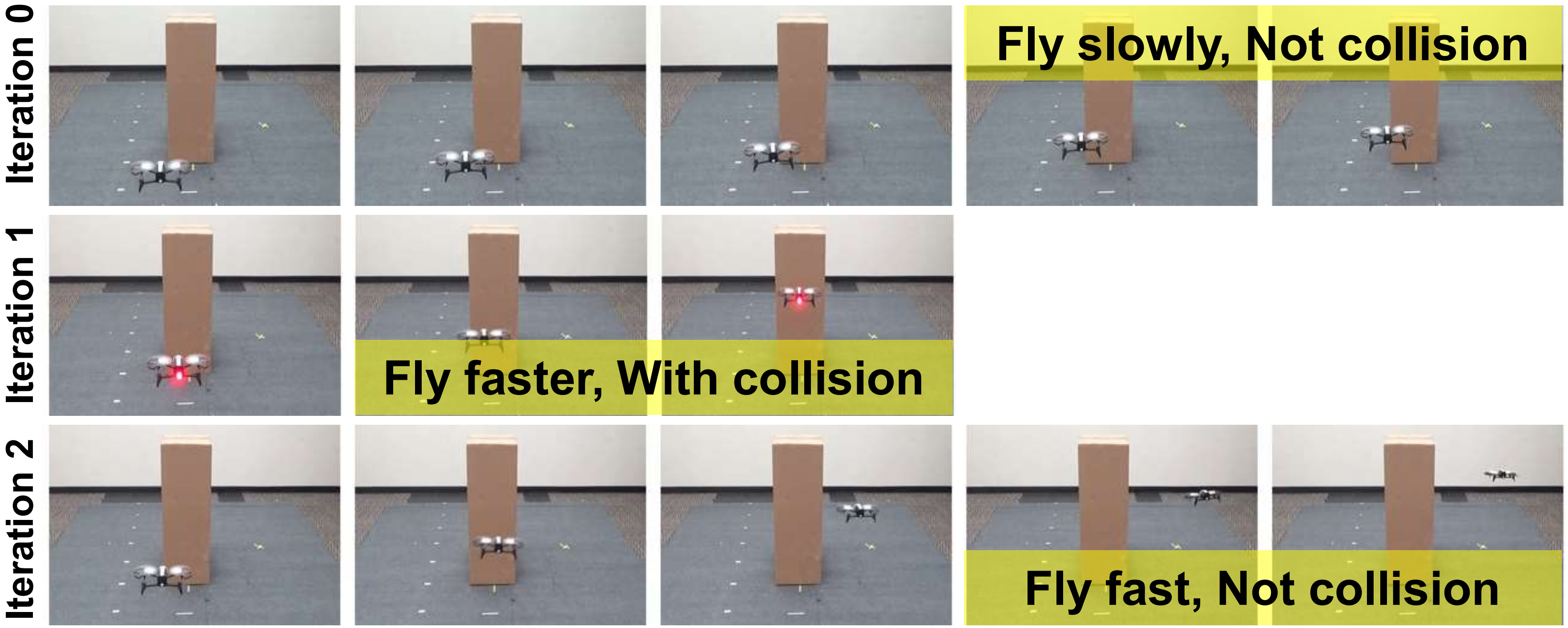
- ○ **Obstacle : cylindrical obstacle**
- ○ **Results**



- λ_{COLL} : non-negative user-defined scalar that weights the relative importance of C_{COLL} versus C_{TASK} / collision cost
- λ_{STD} : non-negative user-defined scalar
- λ_{CONST} : non-negative user-defined scalar replaces the λ_{STD}

Results (3)

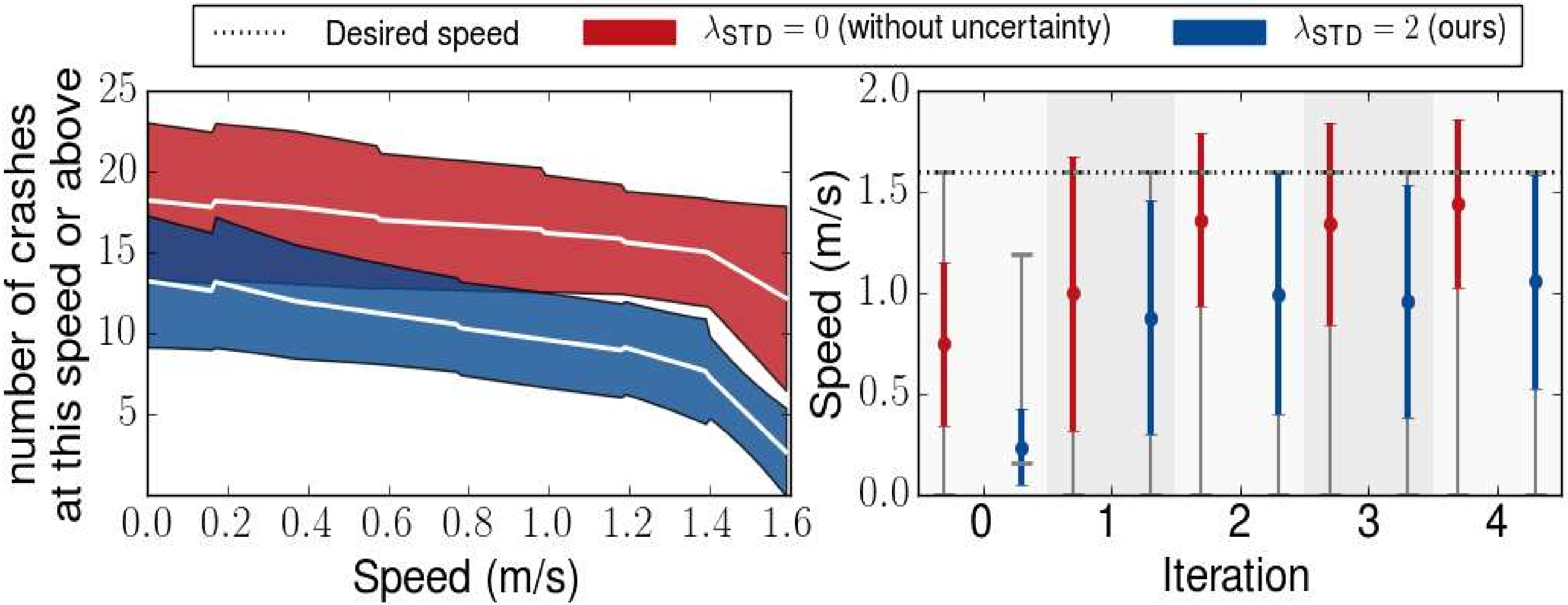
- Real-world quadrotor
  - Obstacle : rectangular obstacle
  - Results



Results (3)

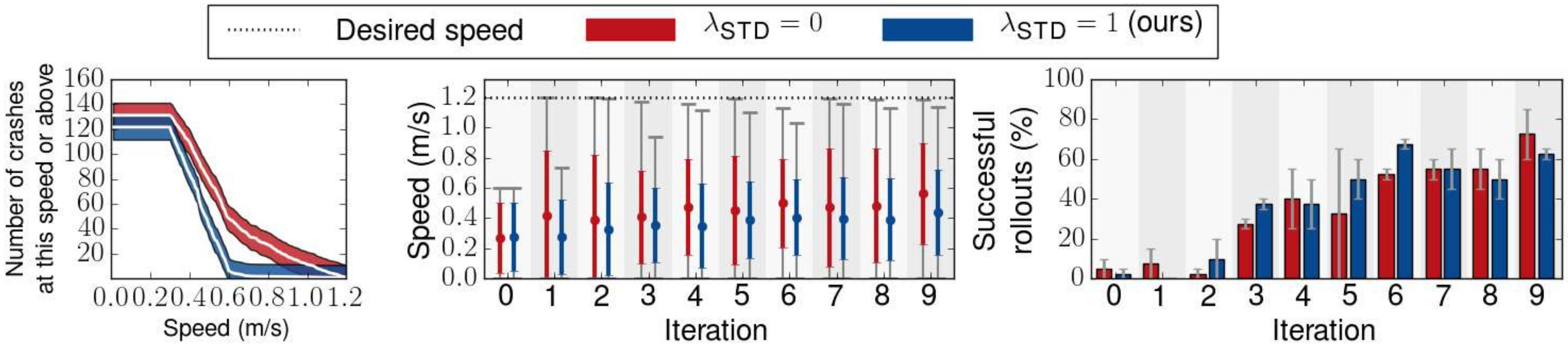
□ Real-world quadrotor

- ○ **Results**



Results (4)

- Real-world RC car
  - Obstacle : Circular cone obstacle
  - Results



Results (5)

Uncertainty-Aware Reinforcement Learning
for Collision Avoidance

Discussion

□ The advantages of this approach

- By directly estimating model uncertainty, we do not rely on a discriminative safety estimator
- Does not assume the existence of a manually designed safety control, but instead naturally reverts to more cautious exploratory behavior in the presence of uncertainty.

Summary and Q&A

- **Summary**
  - **Model-based combined perception and control method for learning obstacle avoidance**
  - **Predict the probability of collision conditioned on raw sensory inputs and a sequence of actions**
  - **This approach is safer compared to methods without uncertainty estimates in experiments**
- **Any Question?**