Interpretable Trade-offs Between Robot Task Accuracy and Compute Efficiency

Authors: Bineet Ghosh[1], Sandeep Chinchali[2], Parasara Sridhar Duggirala[1]

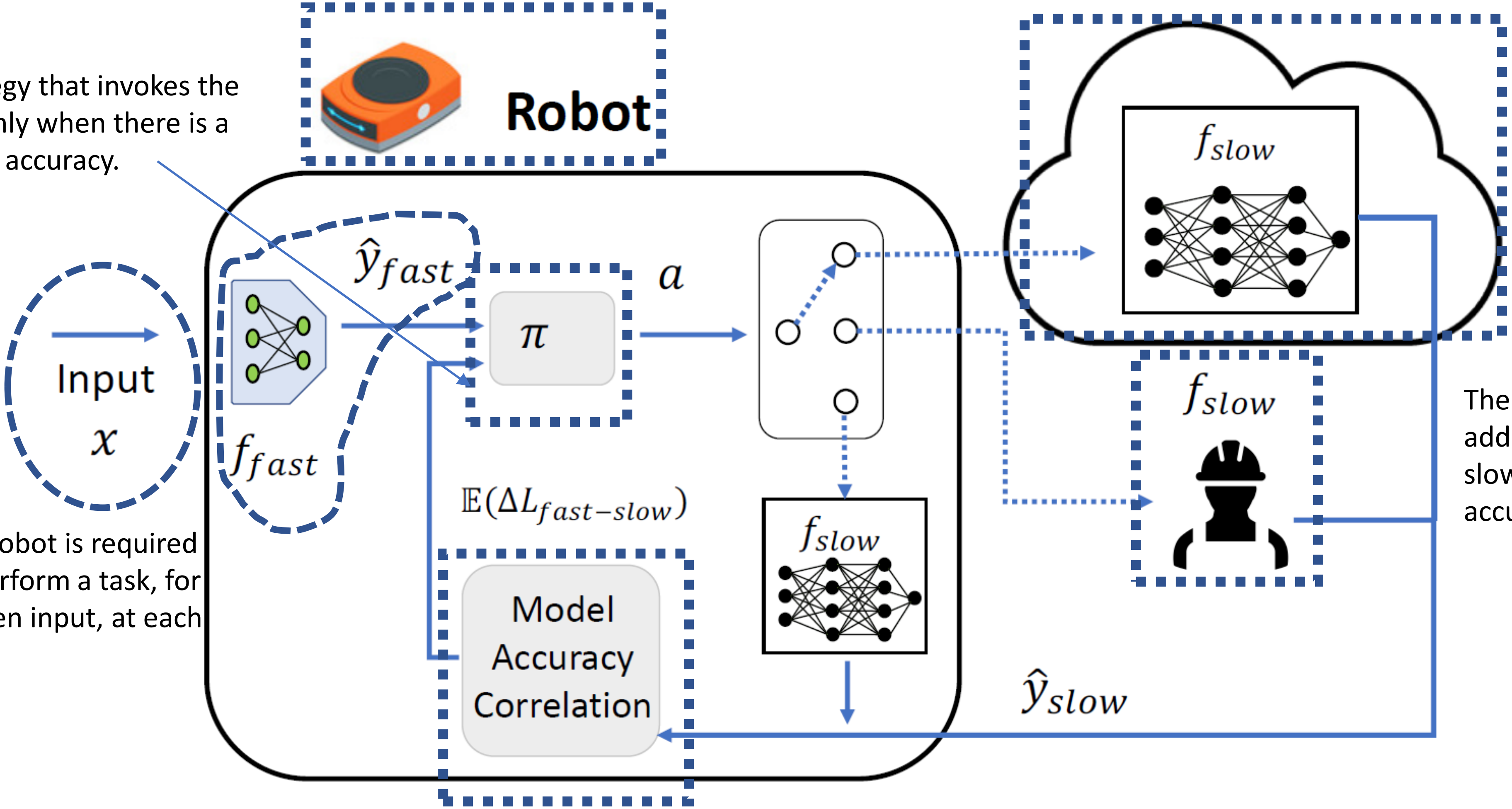
IROS 2021

[1]**Bineet Ghosh** and **Parasara Sridhar Duggirala** are with the Department of Computer Science, The University of North Carolina at Chapel Hill, USA
{bineet,psd}@cs.unc.edu

[2]**Sandeep Chinchali** is with the Department of Electrical and Computer Engineering, The University of Texas at Austin, USA
sandeepc@utexas.edu

Introduction: Model Selection Problem

Propose a strategy that invokes the slower model only when there is a provable gain in accuracy.



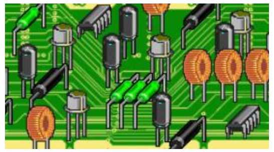
The robot is required to perform a task, for a given input, at each step.

The robot has additional access to a slower, but highly accurate, model.

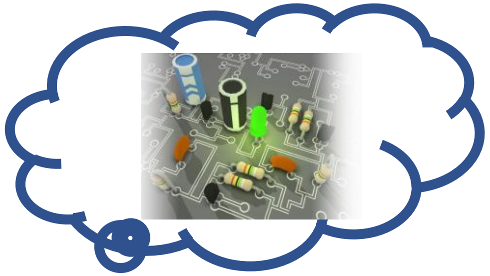
The robot is aware of the accuracy correlation between the models

Introduction: Model Selection Problem

Always available

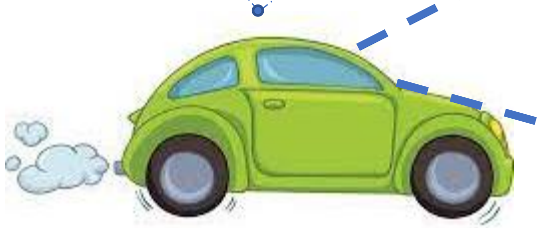


Local: Fast, and less accurate



Cloud: Slow, but highly accurate

Invoke the cloud or not?

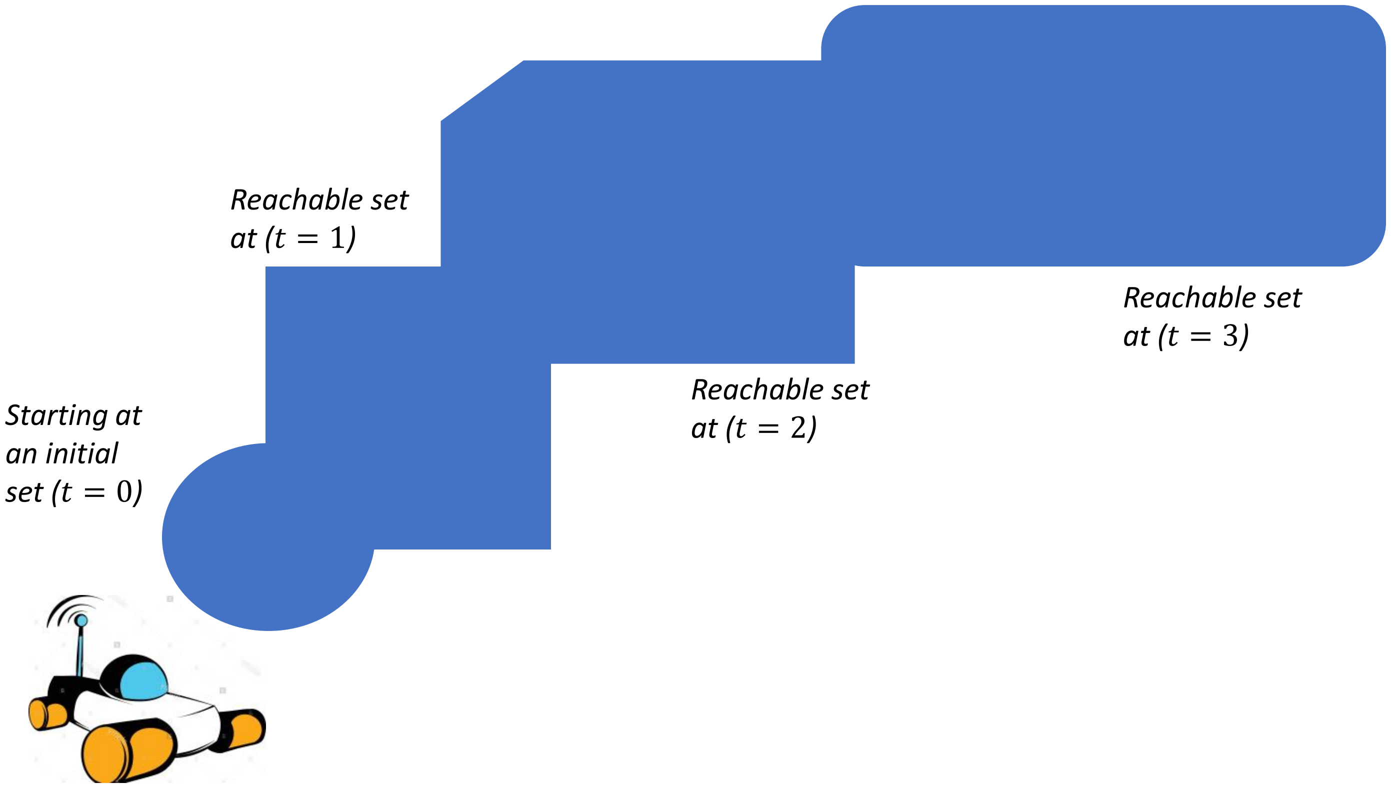


A robot needs to **perform a task** based on its **perception**, *such as autonomous driving*

Model Selection Problem

Motivation: Safe Navigation

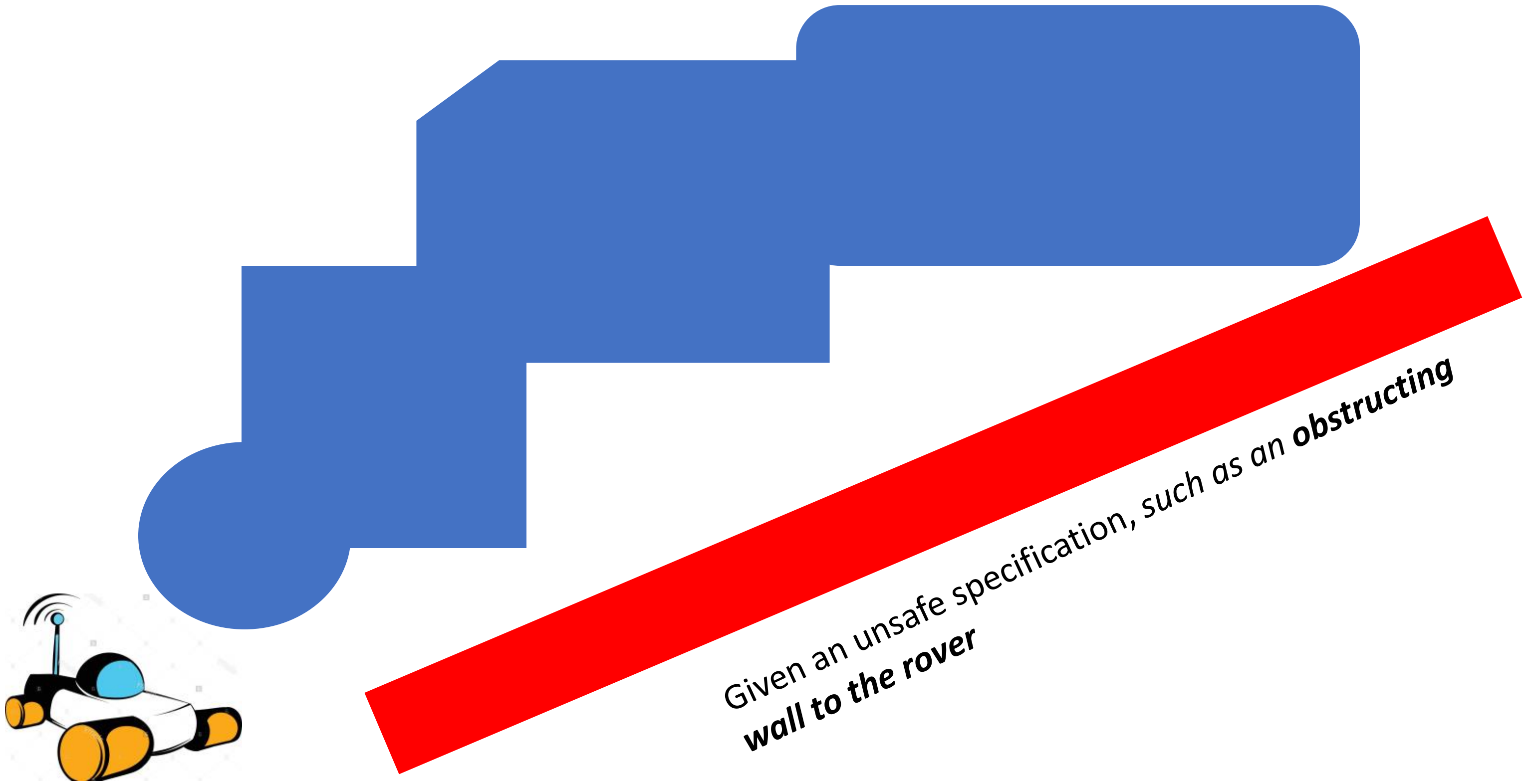
- Safe navigation using ***Reachable Sets***.
- What are ***Reachable Sets***?



Motivation: Safe Navigation

- When is the navigation **safe**?

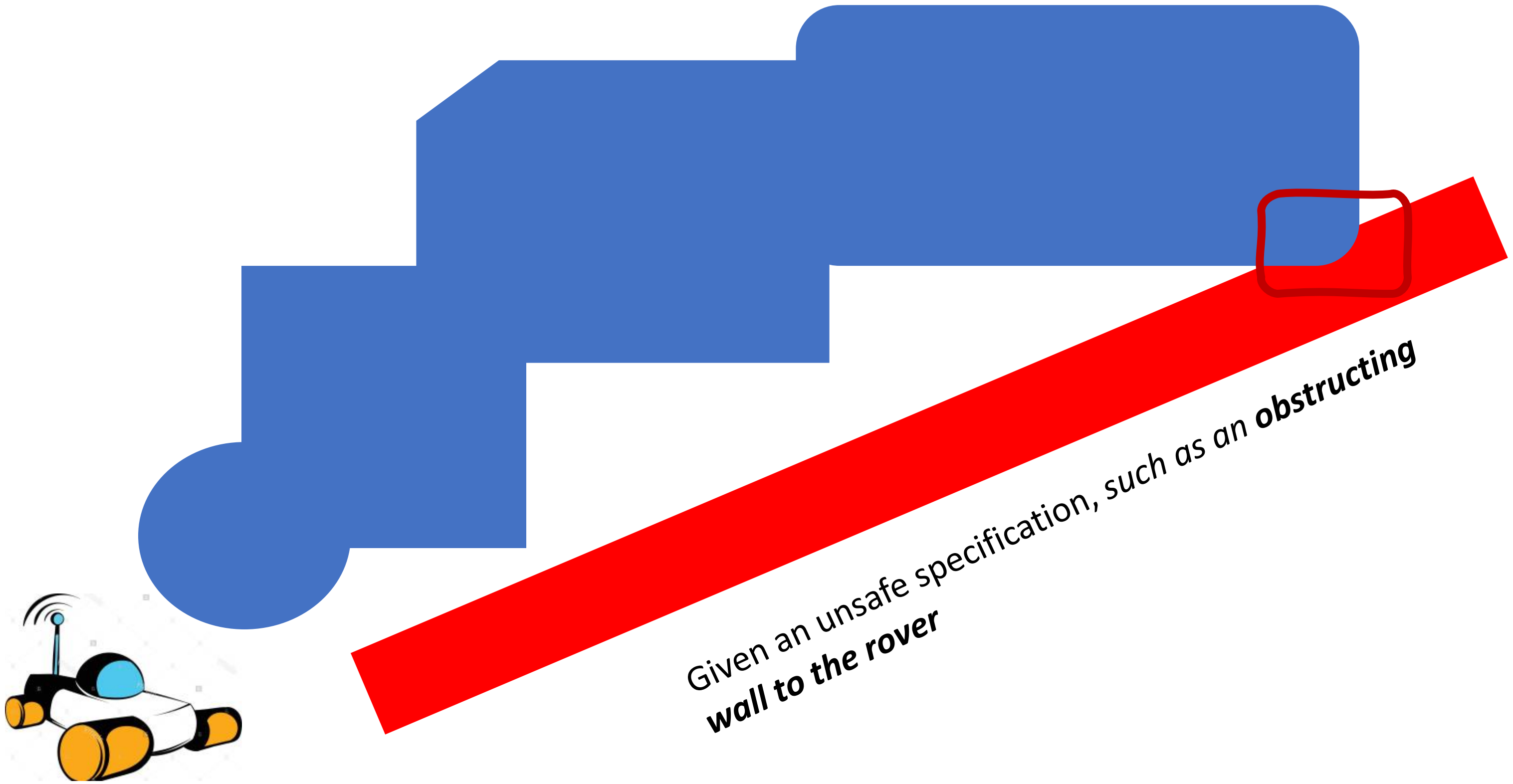
Safe: Reachable set **does not intersect** with unsafe set.



Motivation: Safe Navigation

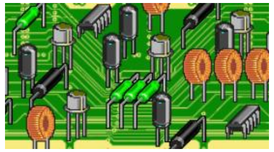
- When is the navigation **safe**?

Unsafe: Reachable set **intersects** with unsafe set.

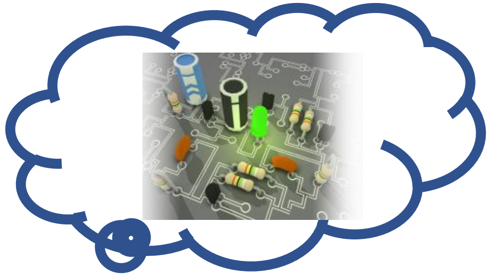


Motivation: Safe Navigation

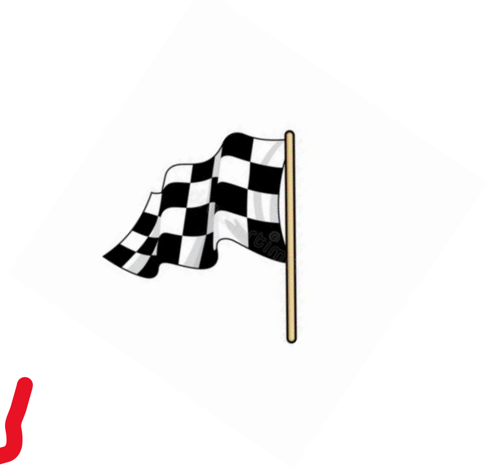
Always available

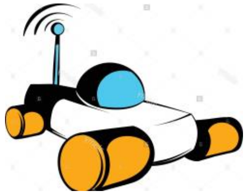


Local: Fast, and low confidence reachable sets



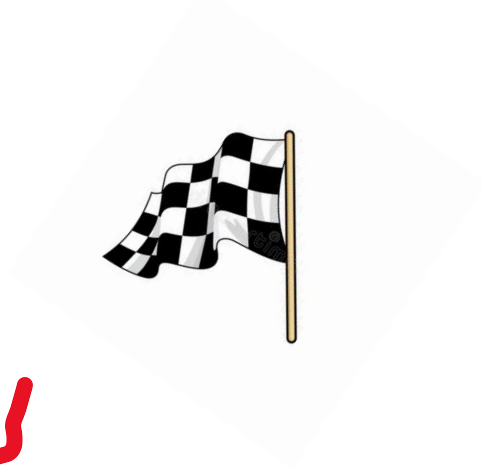
Cloud: Slow, but high confidence reachable sets



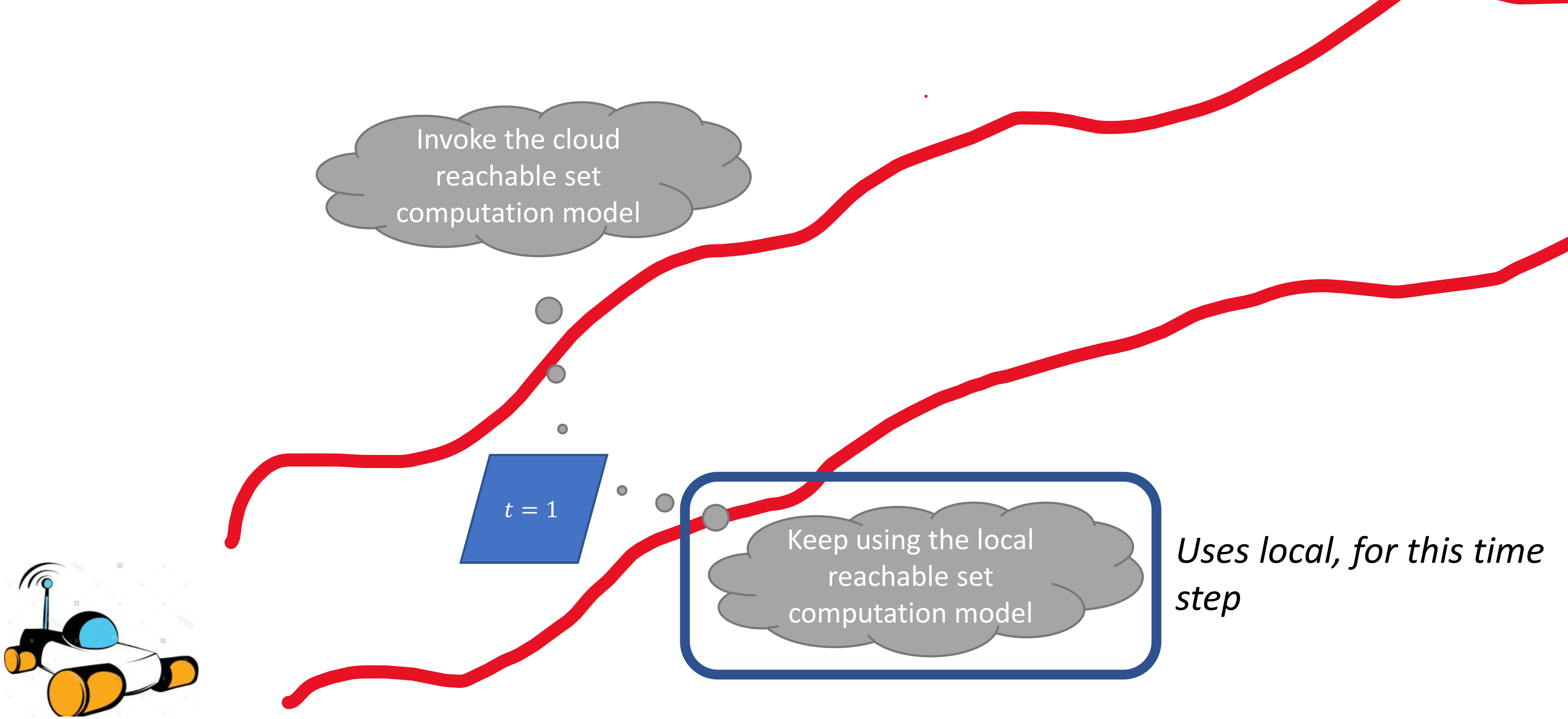


Task: Reach the goal, avoiding obstacles.

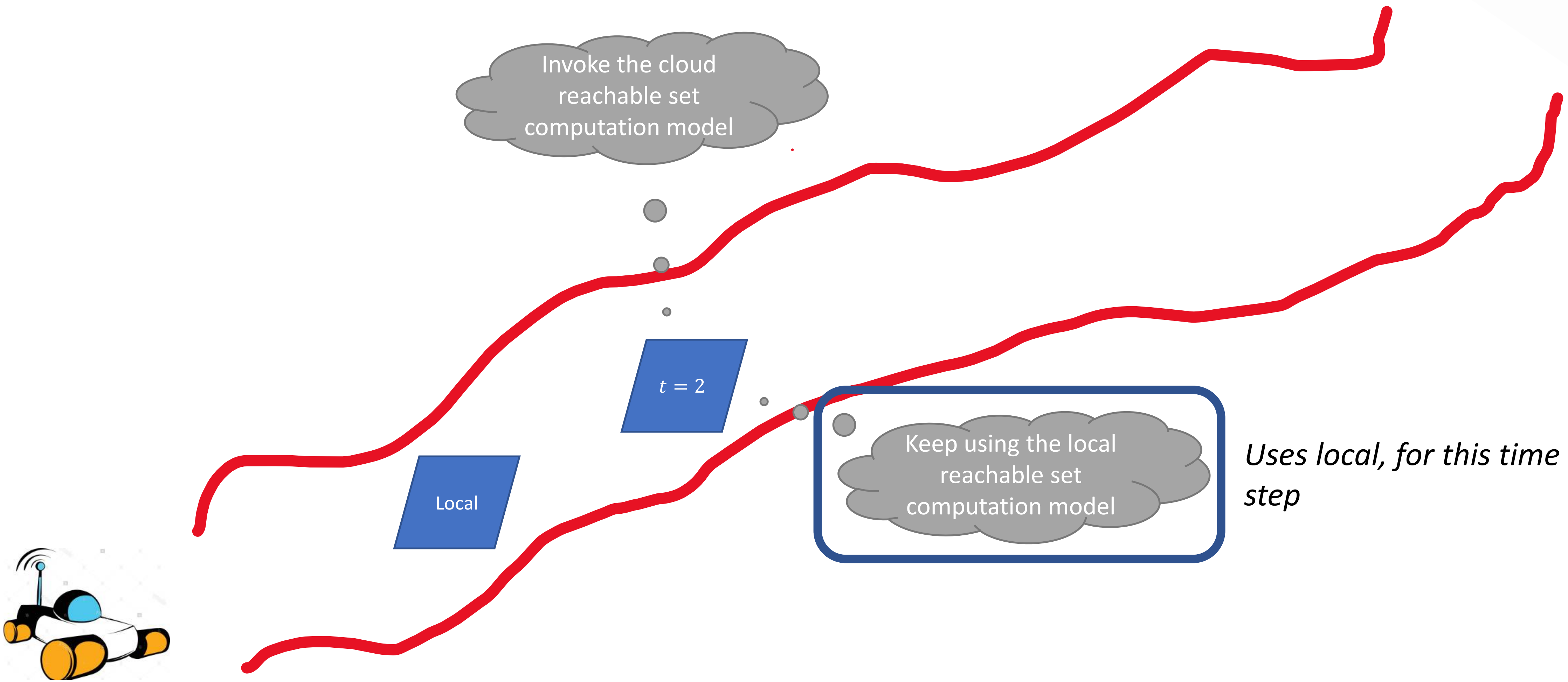
Motivation: Safe Navigation



- At each step: the robot needs to **check safety**.
- **Check safety**: compute reachable set and check for intersection.



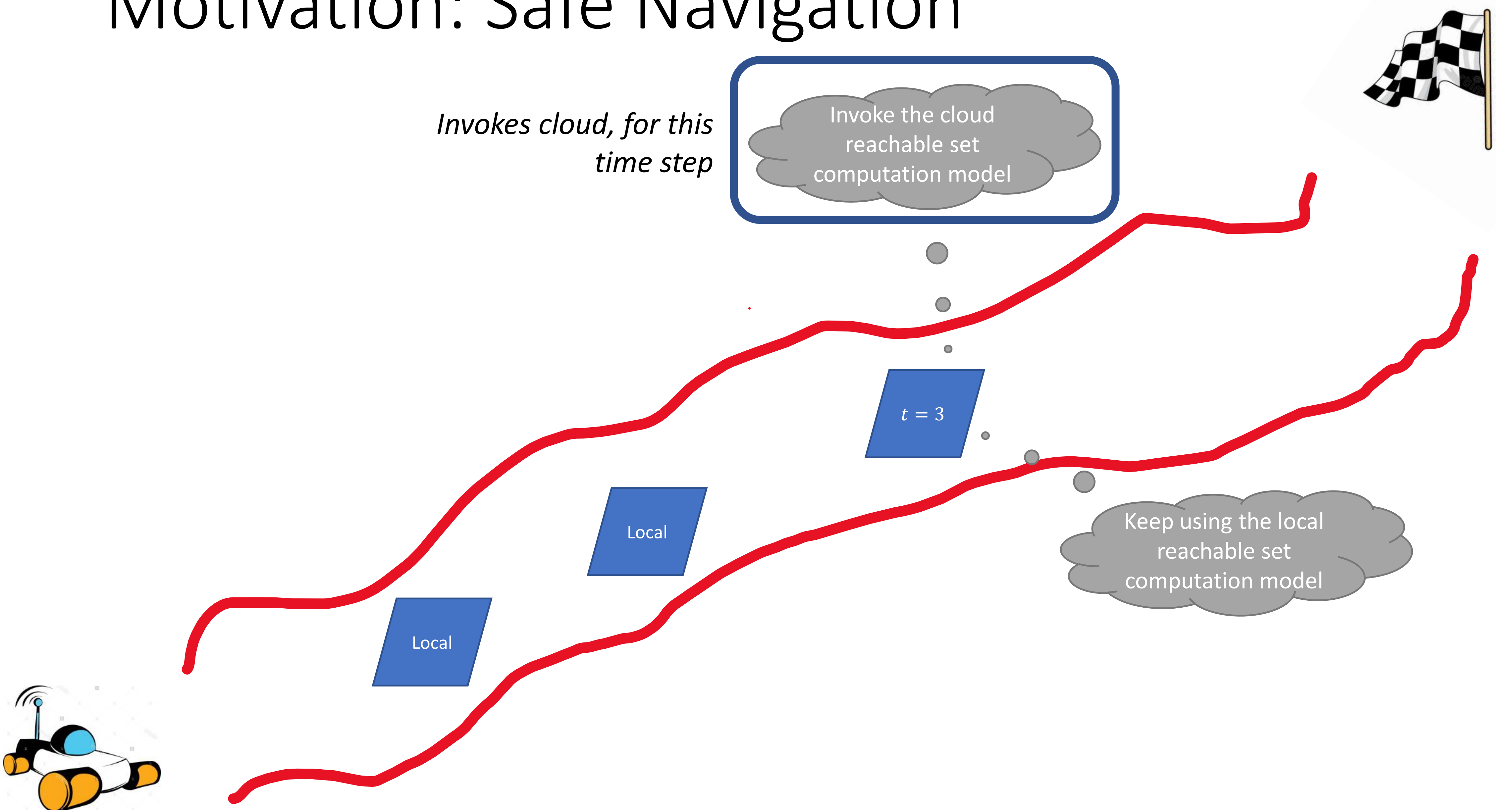
Motivation: Safe Navigation



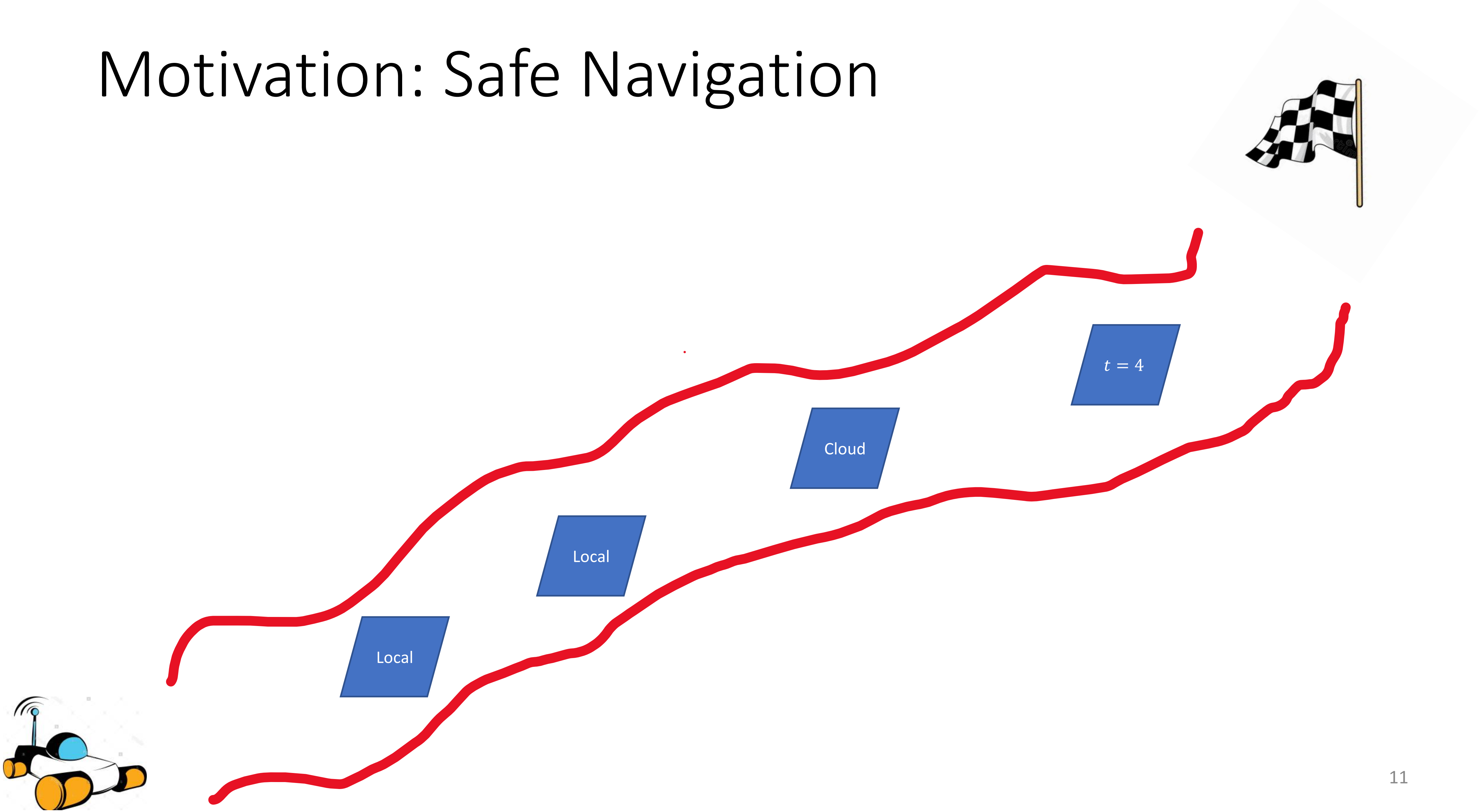
Uses local, for this time step

Motivation: Safe Navigation

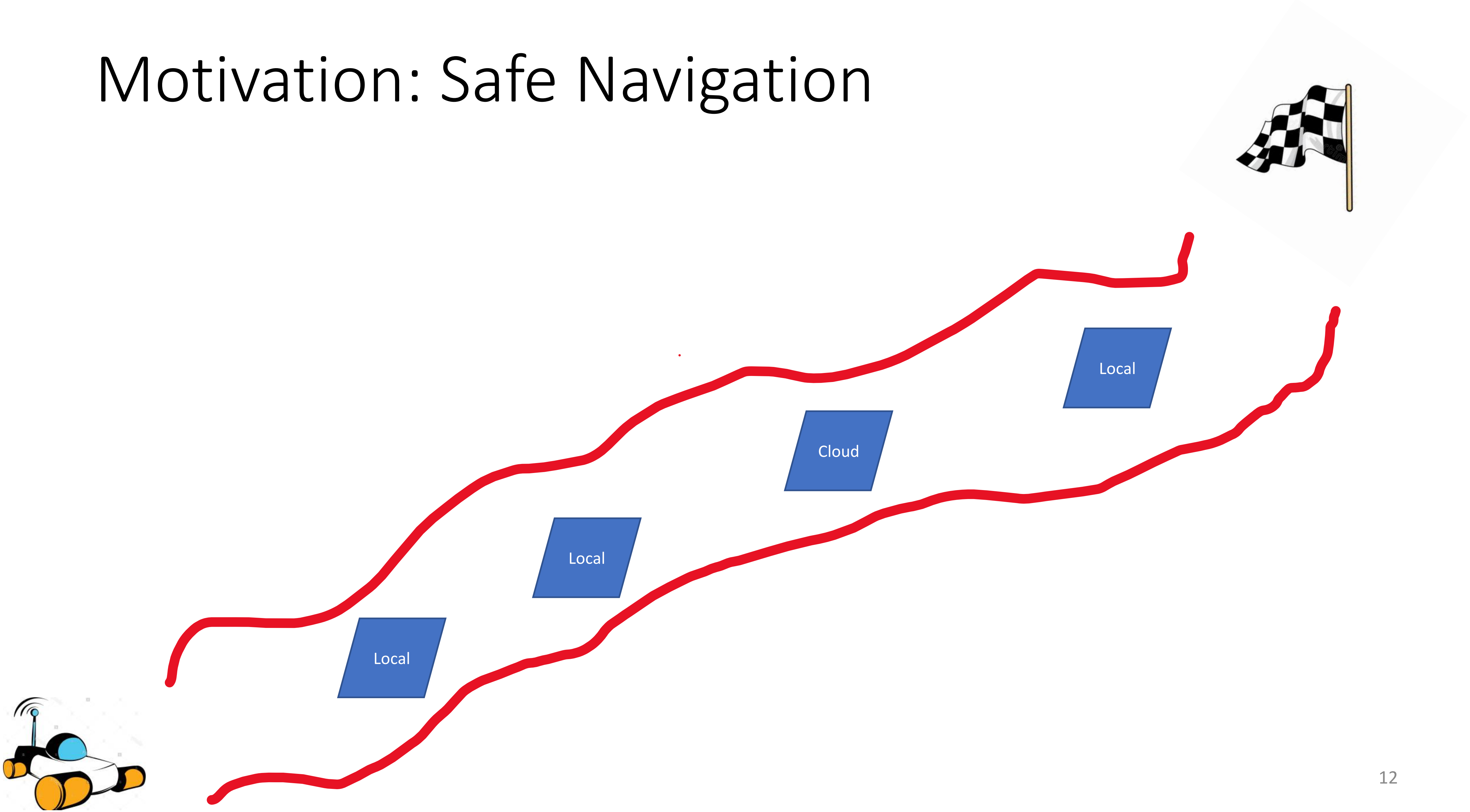
Invokes cloud, for this time step



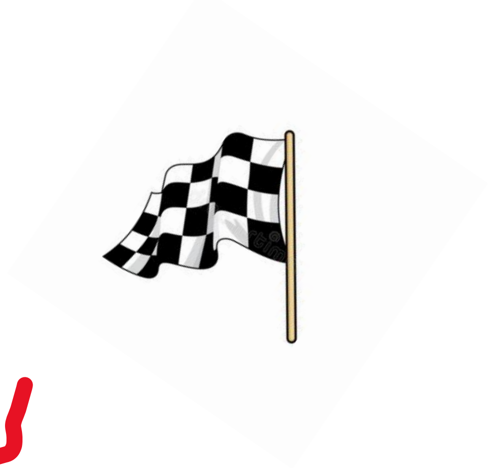
Motivation: Safe Navigation



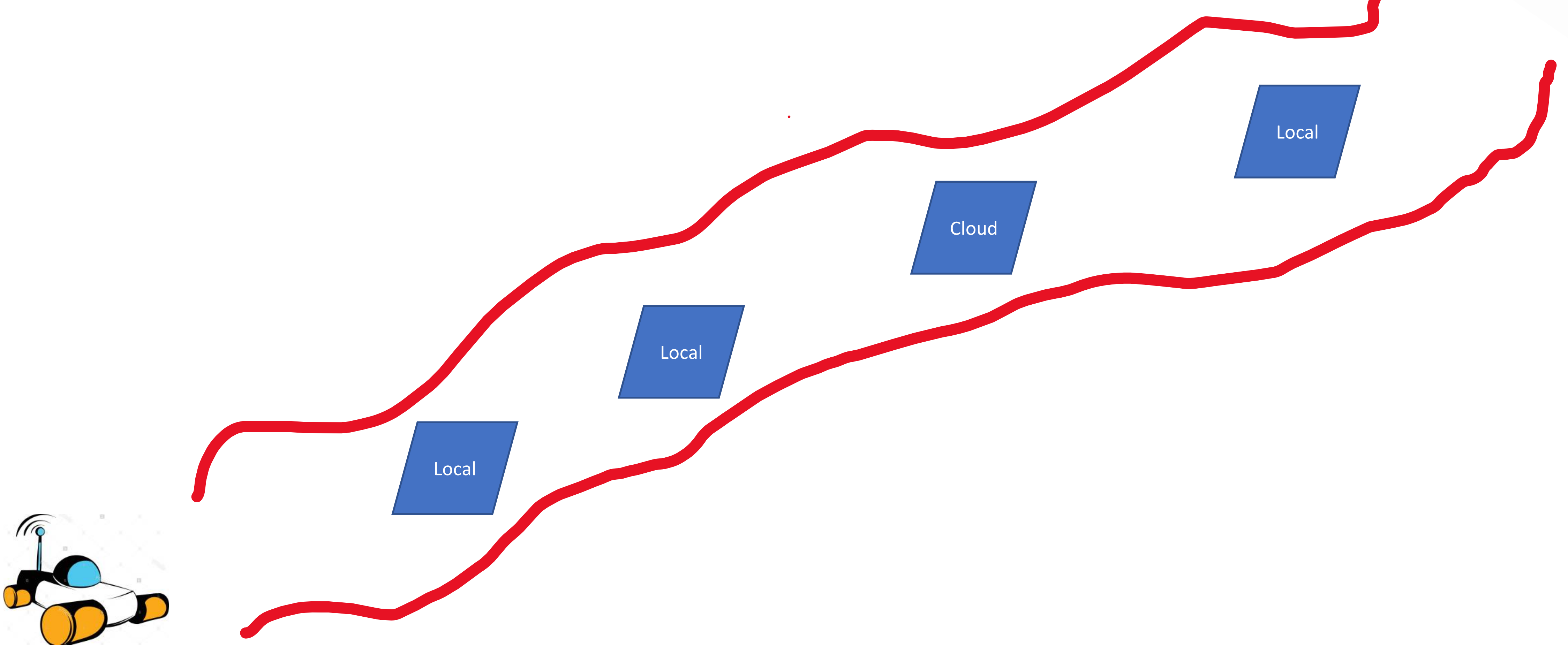
Motivation: Safe Navigation



Motivation: Safe Navigation



Model Selection Sequence: *Local, Local, <u>Cloud</u>, Local*



Model Features: Loss and Cost

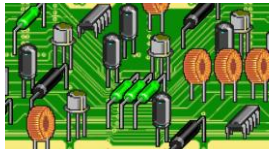
- Define: Loss associated with the models.

- Define: Cost associated with the models.

Optimality: Terms

- **Loss** *(conversely, accuracy)*: *Depends on the computation model.*
  - Linear Regression, DNN: *L1, L2, etc.*
  - Safe Navigation using reachable sets: *Confidence, Binary Loss, etc.*

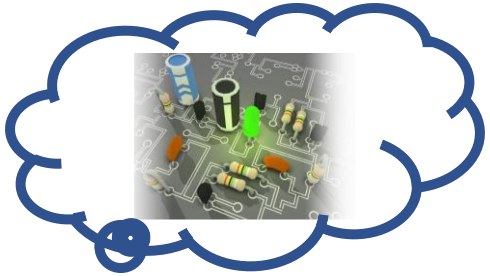
- **Cost**: *Depending on the task involved.*
  - Examples: *compute time of the model (local/cloud), etc.*



Local: Fast, and less accurate

$\geq_{loss}$

$\leq_{cost}$



Cloud: Slow, but highly accurate

Contribution of this Paper

- **Provably Optimal** model selection strategy (or sequence).

- Next: Formally define **Optimality** in terms of *Loss and Cost*.

Optimality: Reward

- Reward: To define optimality formally, we define $Reward_t$, **<u>at each step t</u>**, as:

- $Reward_t = -\alpha \cdot Loss_t - \beta \cdot Cost_t$

α, β: User Given

Penalize the loss at the given step — based on the selected model (local/cloud)

Penalize the cost incurred at the given step — based on the selected model

Optimality: Reward

- Reward of Cloud and Local:

  - $Reward_t^{Cloud} = -\,\alpha \cdot Loss_t^{Cloud} - \beta \cdot Cost_t^{Cloud}$
  - $Reward_t^{Local} = -\,\alpha \cdot Loss_t^{Local} - \beta \cdot Cost_t^{Local}$

Optimality: Problem Statement

- Total Reward: Cumulative reward up to time T:

  - $Reward = \sum_{t=1}^{T}(Reward_t)$

- **<u>Objective</u>**: *Compute a model selection strategy – <u>a sequence of Local/Cloud</u> – that maximizes the cumulative reward.*

All possible sequences

`Local, Local, . . ., Local, Local`

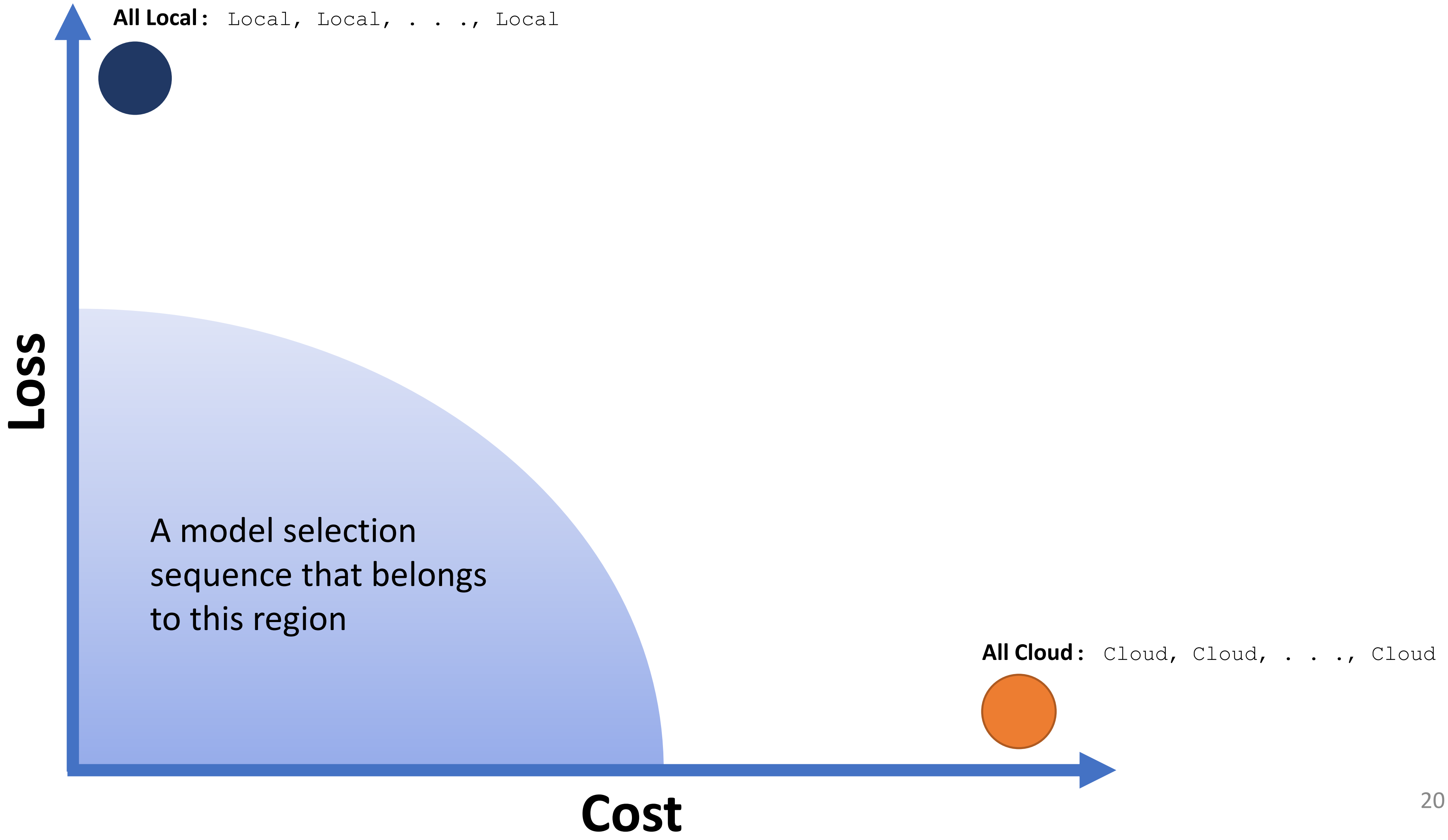
`Local, Local, . . ., Local, Cloud`

`Local, Cloud, . . ., Cloud, Local`

`Cloud, Cloud, . . ., Cloud, Cloud`

*Compute a sequence – **without exploring all sequences** – that maximizes the cumulative reward*

Optimality: Problem Statement Illustration



Crux of the Solution: Intuition

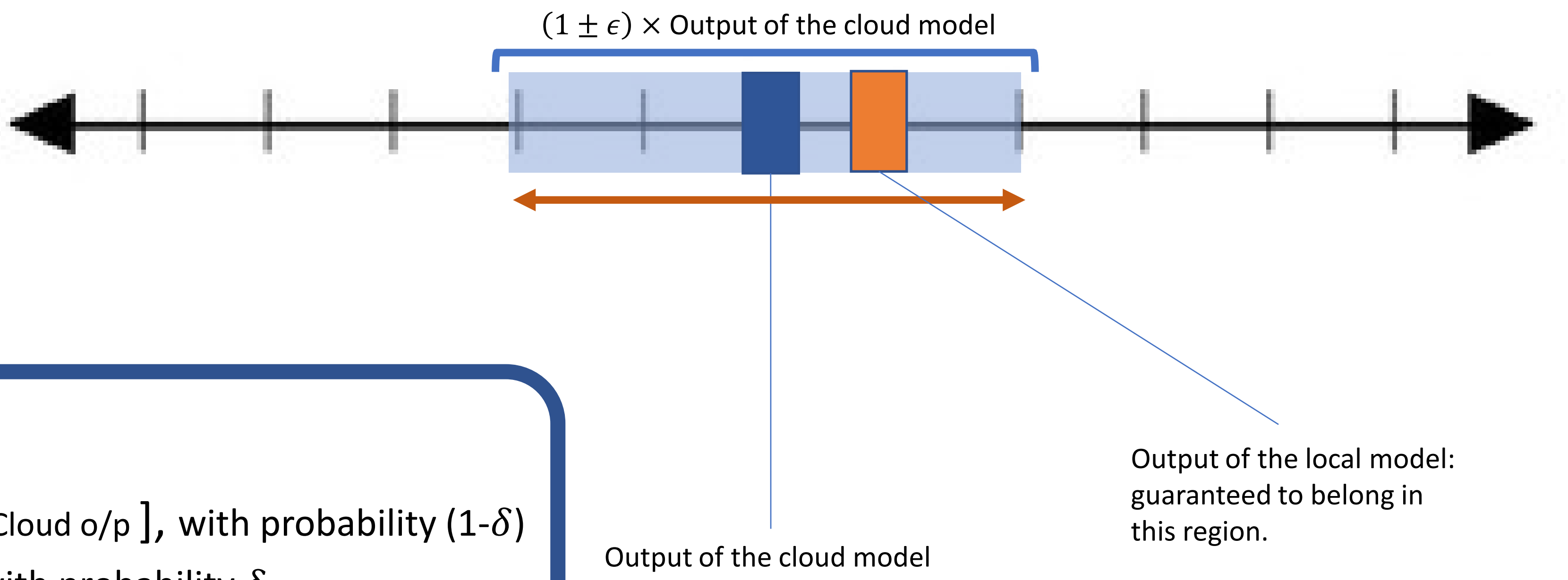
- Intuitively, invoke the cloud model **<u>if and only if</u>**:
  - $Reward_t^{Cloud} > Reward_t^{Local}$

Computing this, ***without invoking the cloud***, is the main challenge

Always available to the robot

Crux of the Solution: Model Relationship

- Leverage the relationship – often statistical – between the local and the cloud models output.

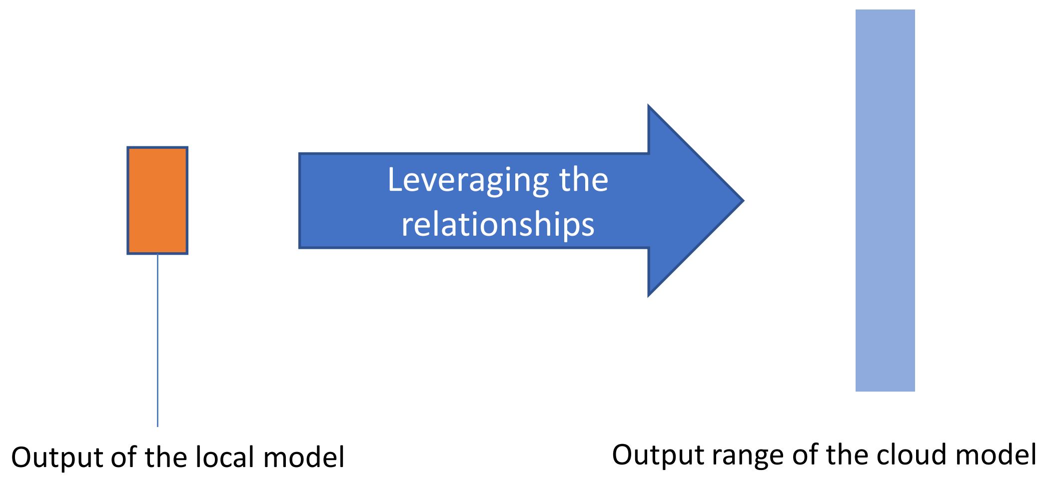


More Precisely:

- Local o/p $\in$ [$(1-\epsilon)\cdot$ Cloud o/p , $(1+\epsilon)\cdot$ Cloud o/p], with probability (1-δ)
- Local o/p $\in$ [Cloud o/p - $\frac{M}{2}$, Cloud o/p - $\frac{M}{2}$], with probability δ

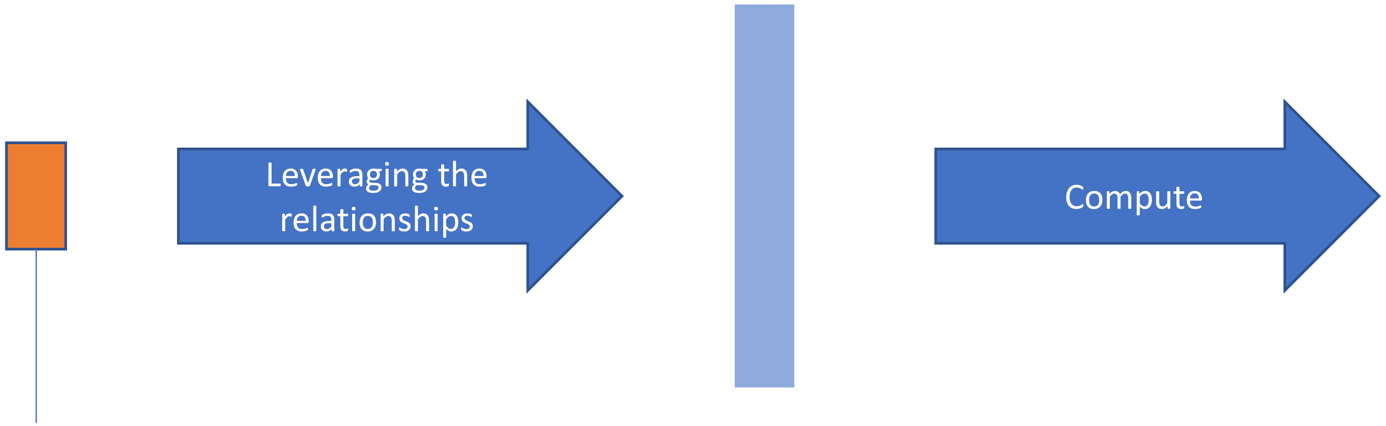
Crux of the Solution: Guess the Cloud

- Robot computes the output range of the Cloud – without invoking it – from the output of local model.



Crux of the Solution: Guess the Cloud Reward

- Robot computes the output range of the Cloud – without invoking it – from the output of local model.
- From the computed output range, compute the expected reward obtained by the Cloud.



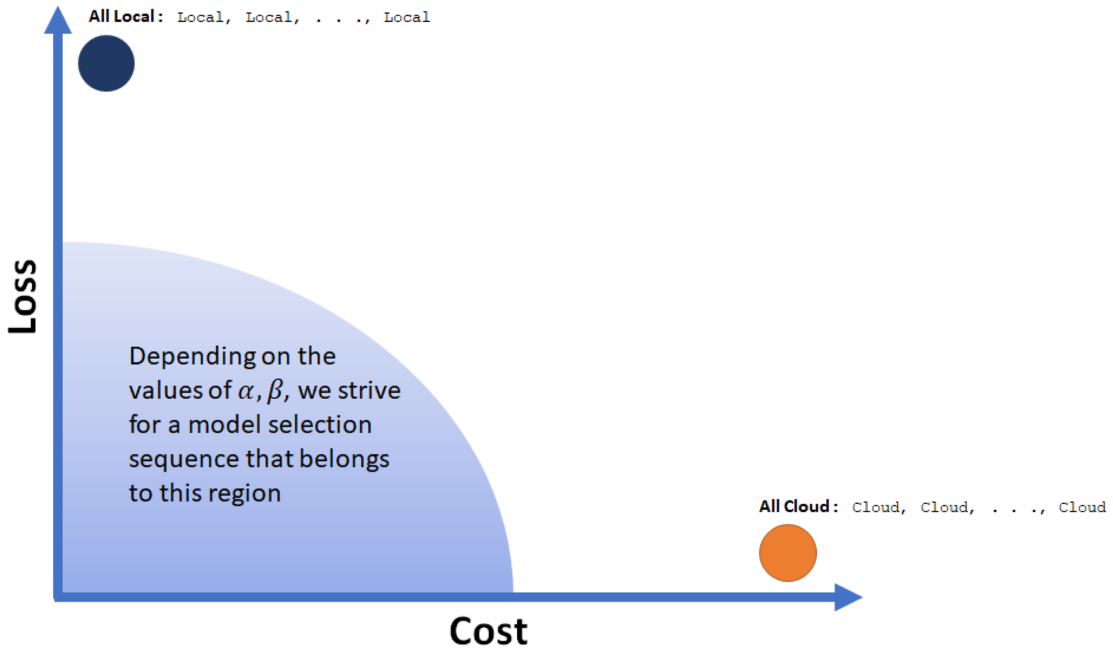
Output of the local model

Output range of the cloud model

Compute the expected $Reward_t^{cloud}$ obtained by the cloud

Crux of the Solution: Model Selection Policy

- At a given time step t, invoke the cloud model if and only if:
  - Expected $Reward_t^{Cloud} > Reward_t^{Local}$



Proof of Optimality: Given in the paper!

Crux of the Solution: Closed Form Solutions

- ***<u>Closed form solutions</u>*** to the model selection policy – $Reward_t^{Cloud} > Reward_t^{Local}$ – for the following cases are given in the paper:

  - Both models (Local/Cloud) are ***Linear Regression***, with varying cost and loss.

  - Both models are ***Deep Neural Network (DNN)***, with varying cost and loss.

  - ***Safe Navigation with Reachable Sets***, where the models compute reachable sets with varying cost and <u>confidence</u> (equivalently, loss).

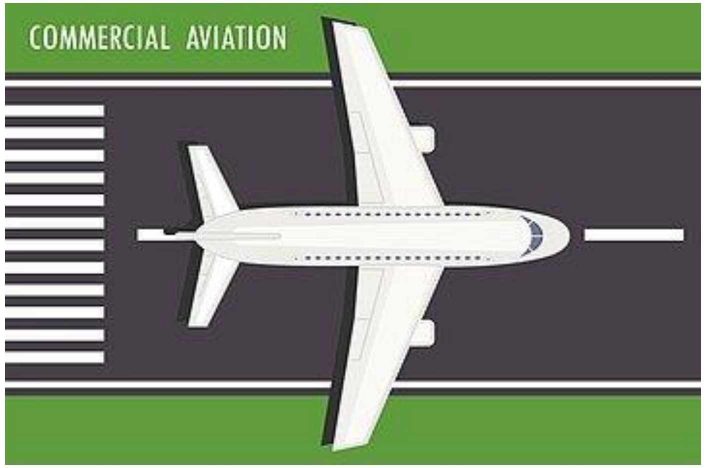
Evaluation

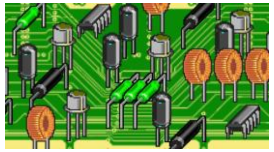
- Demonstrate on following cases:
  - DNN: *Aircraft Taxiing*
  - Safe Navigation with Reachable Sets: *Navigation of a simulated Mars Rover (with uncertainty in the yaw angle) on a real Martian Terrain.*

- Against the following Benchmark:
  - All Robot: Local compute model is used for all time steps.
  - All Cloud: Cloud compute model is used for all time steps.
  - Random: A random sequence of model selection.
  - Oracle: Exact cloud model's output is known – <u>Note that this is an unrealizable policy.</u>
  - Our Selector: Model selection policy proposed in this paper.

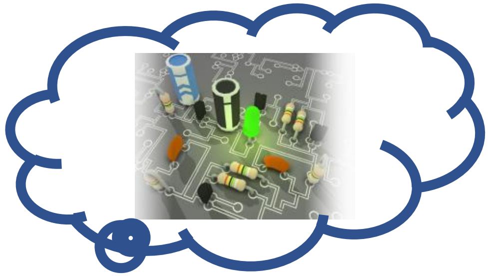
Evaluation: DNN

- Evaluated the model selection policy on *Aircraft Taxiing* – movement of the aircraft with its own power.





Local: Fast, and less accurate

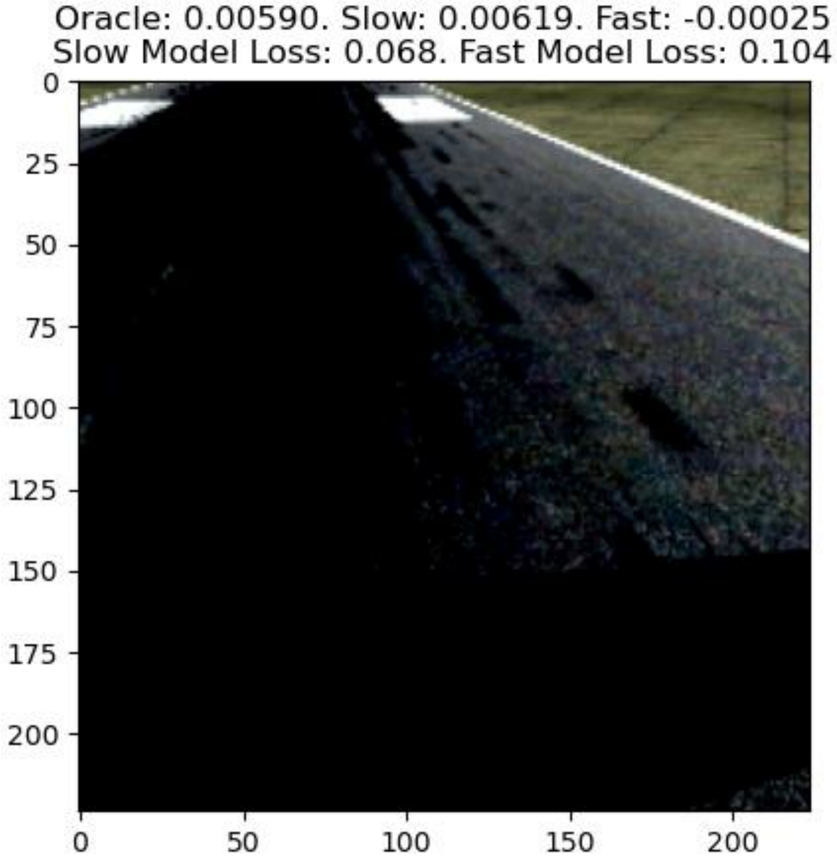


Cloud: Slow, but highly accurate

Rudder Control: a linear combination of the taxiway center and heading angle

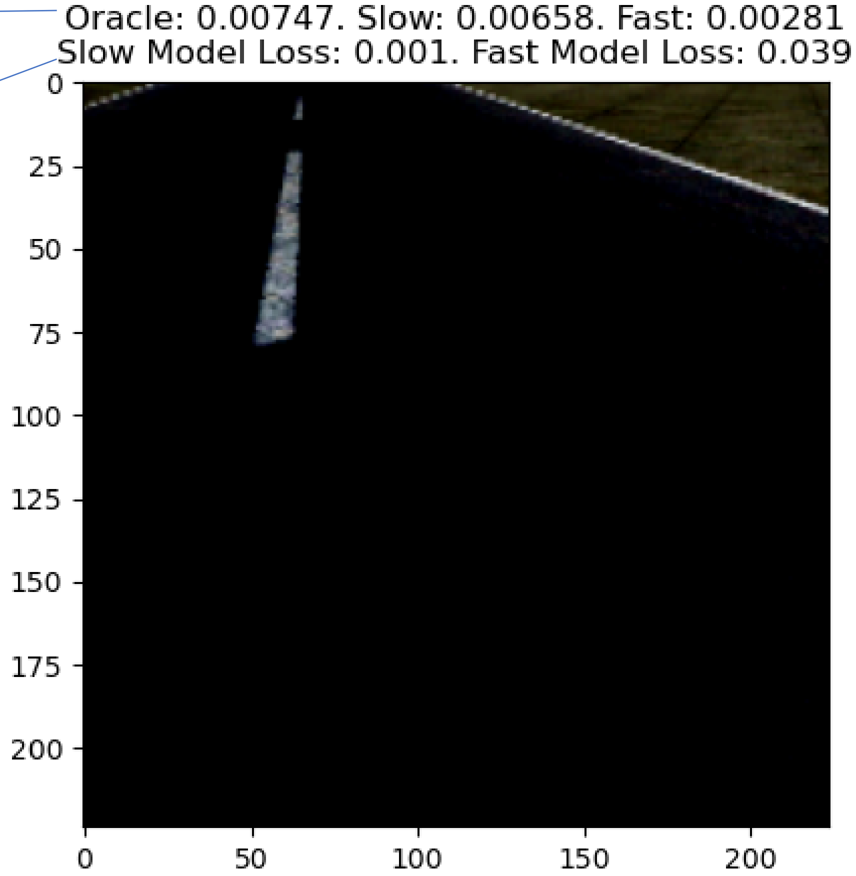
Computed at each time step.

Evaluation: DNN

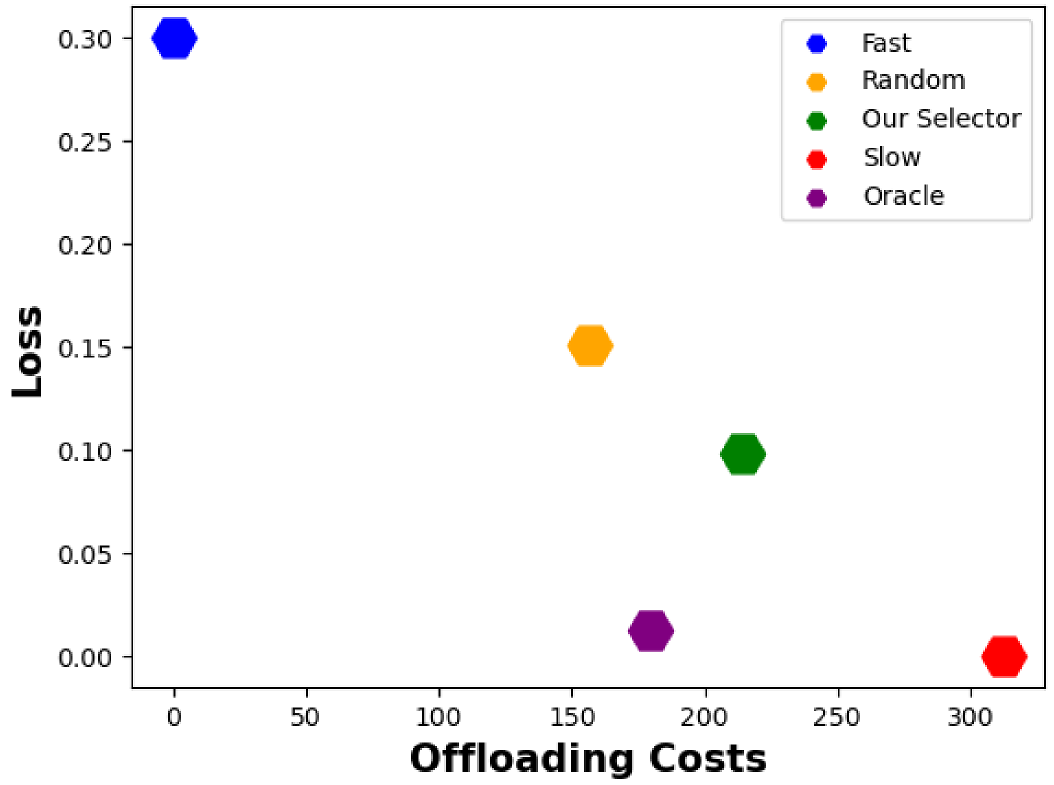


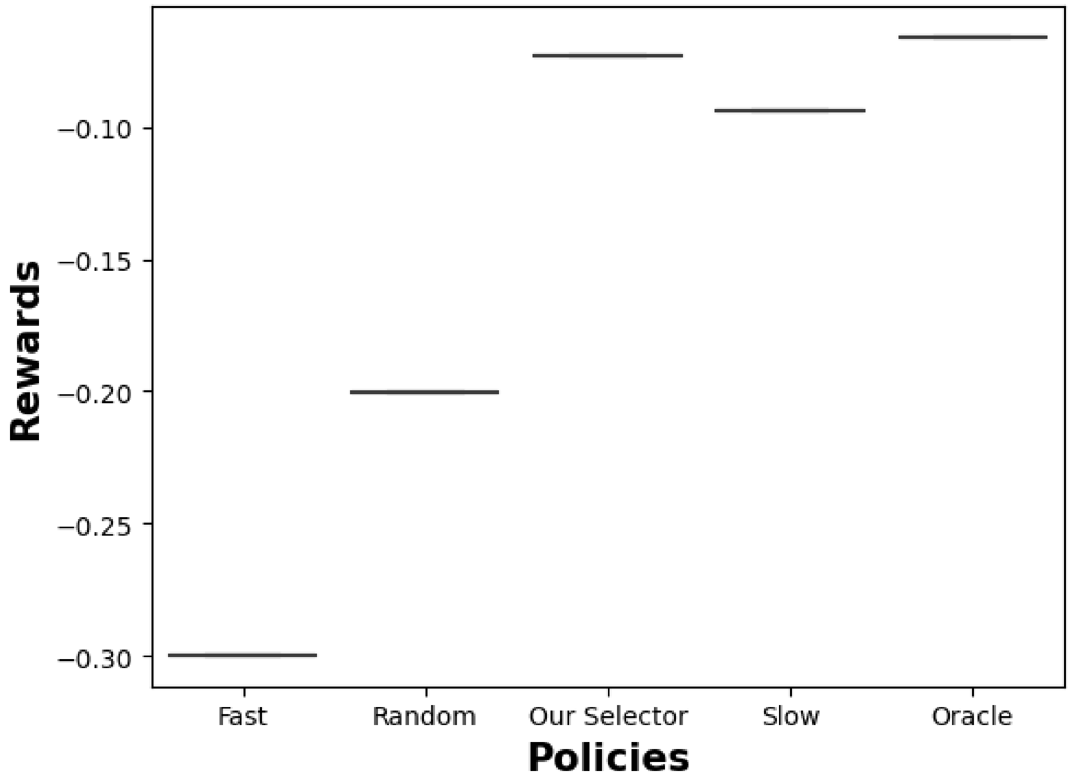
Rudder Controls

Model Losses



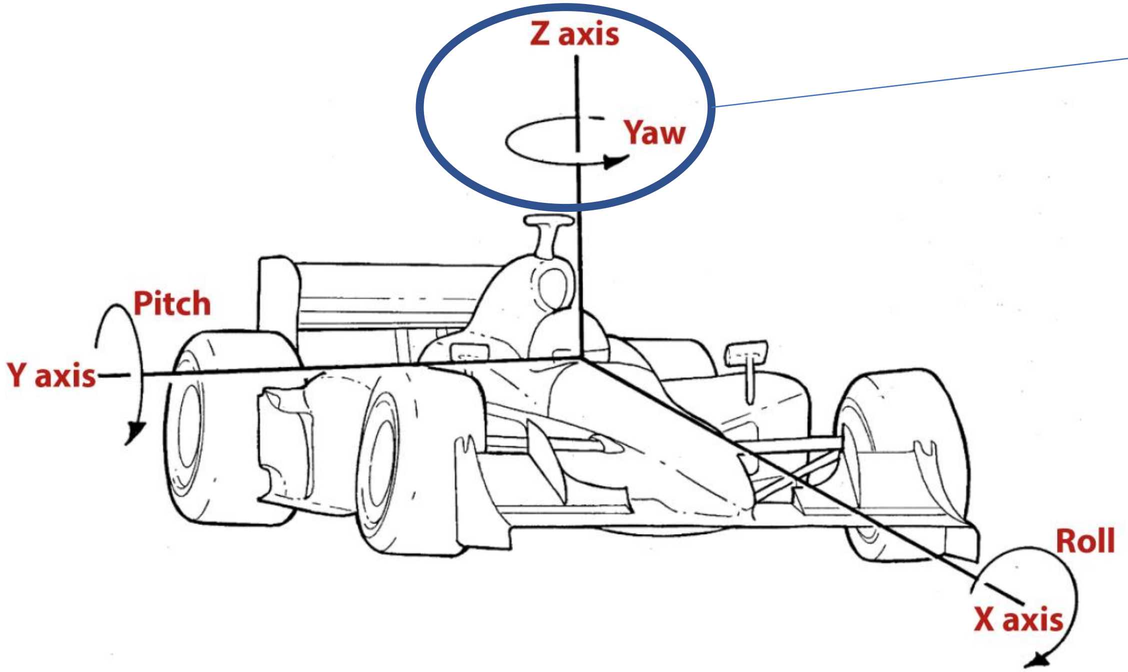
Evaluation: DNN





Evaluation: Safe Navigation with Reachable Sets

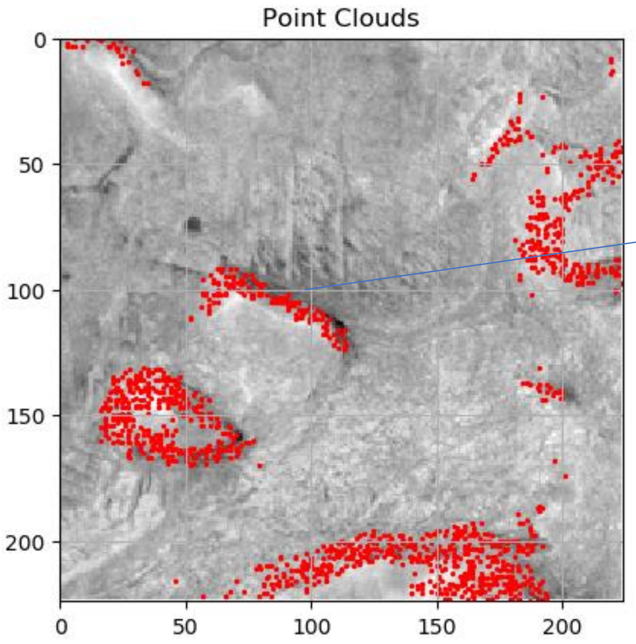
- Evaluated the model selection policy on navigation of a simulated Mars Rover - **with uncertainty in the yaw angle** - on a real Martian Terrain.



The Rover's sensor, responsible for calculating the yaw angle, has an error associated with its reading

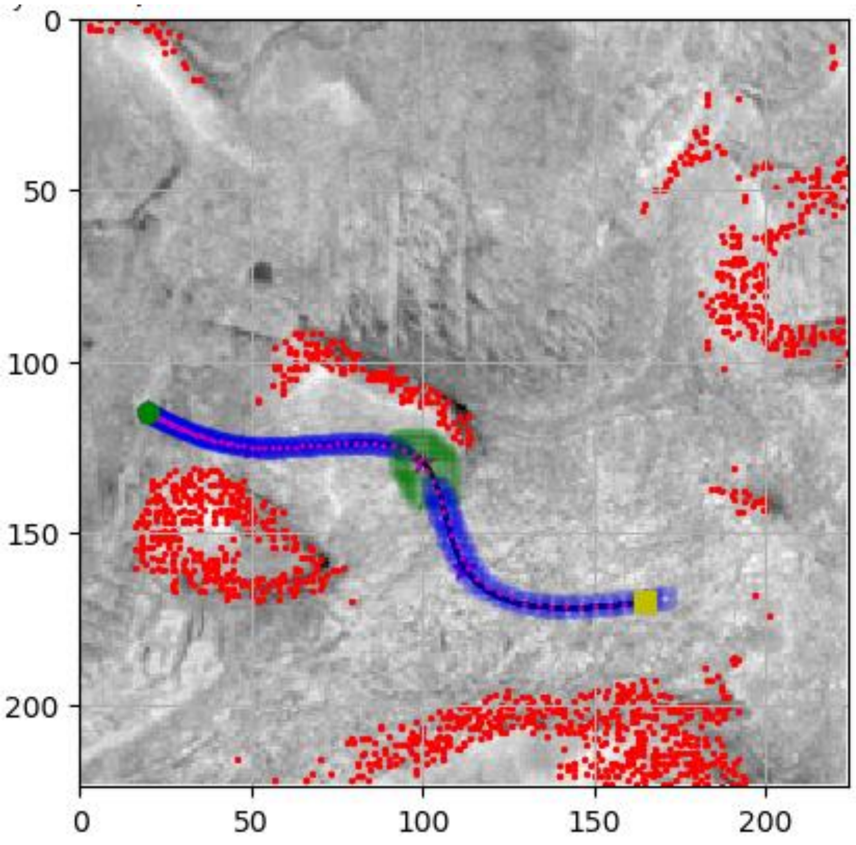
Evaluation: Safe Navigation with Reachable Sets

- Evaluated the model selection policy on navigation of a simulated Mars Rover - with uncertainty in the yaw angle - **on a real Martian Terrain**.



The red point clouds are obstacles indicating regions of high elevation

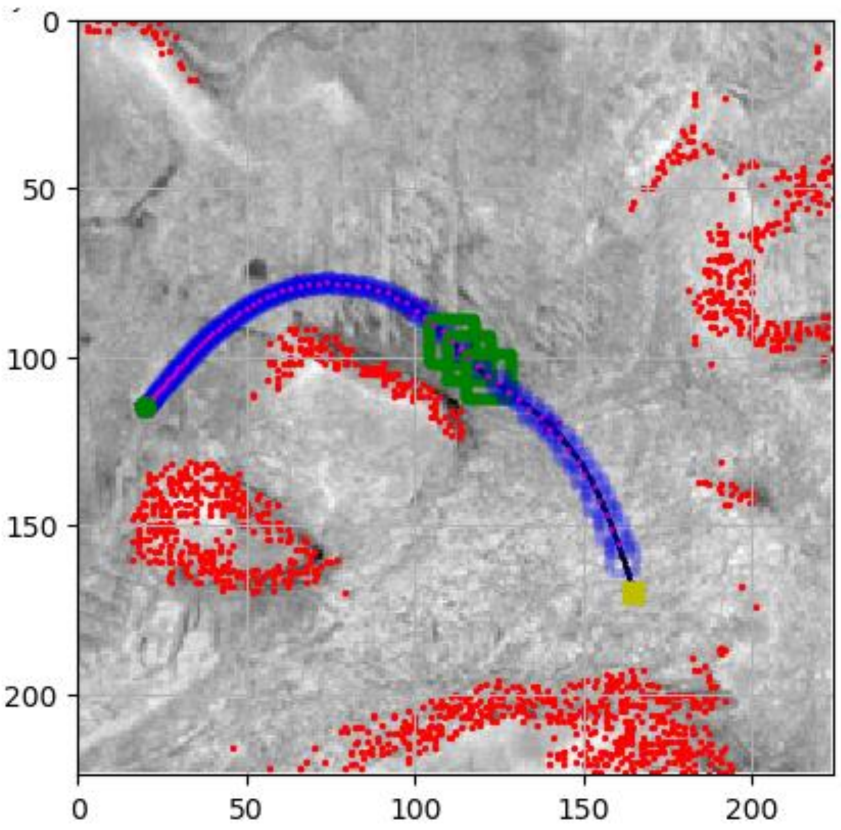
Evaluation: Safe Navigation with Reachable Sets



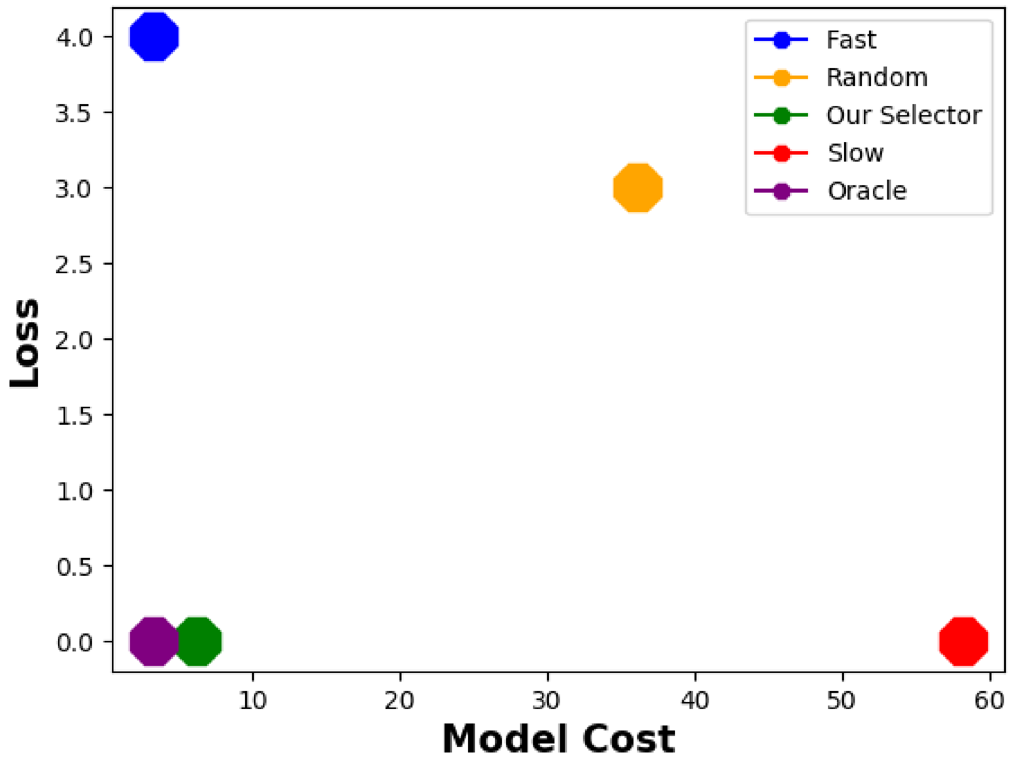
Blue: Low confidence reachable sets obtained from the local model

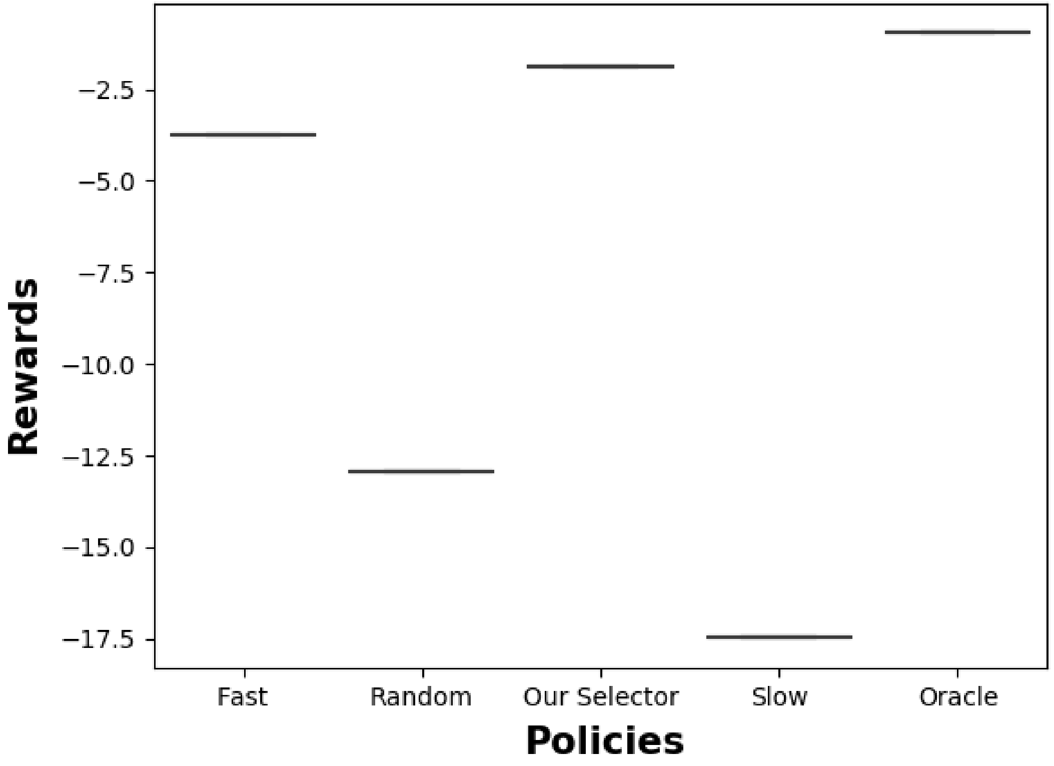
Green: High confidence reachable sets obtained from the cloud model

Insight: Our model selection policy invokes the cloud model only when the Rover is making tricky maneuvers.



Evaluation: Safe Navigation with Reachable Sets





Thank You!