Fast Option Ranking in Autonomous Systems for Criticality Evasion under Uncertainties

Bineet Ghosh[1] Parasara Sridhar Duggirala[2] Samarjit Chakraborty[2]

FDL 2025

[1]**Bineet Ghosh** is with the Department of Computer Science, **The University of Alabama**, USA.
bineet@ua.edu, bineet.cs.ua.edu

[2]**Parasara Sridhar Duggirala** and **Samarjit Chakraborty** is with the Department of Computer Science, **The University of North Carolina at Chapel Hill**, USA.

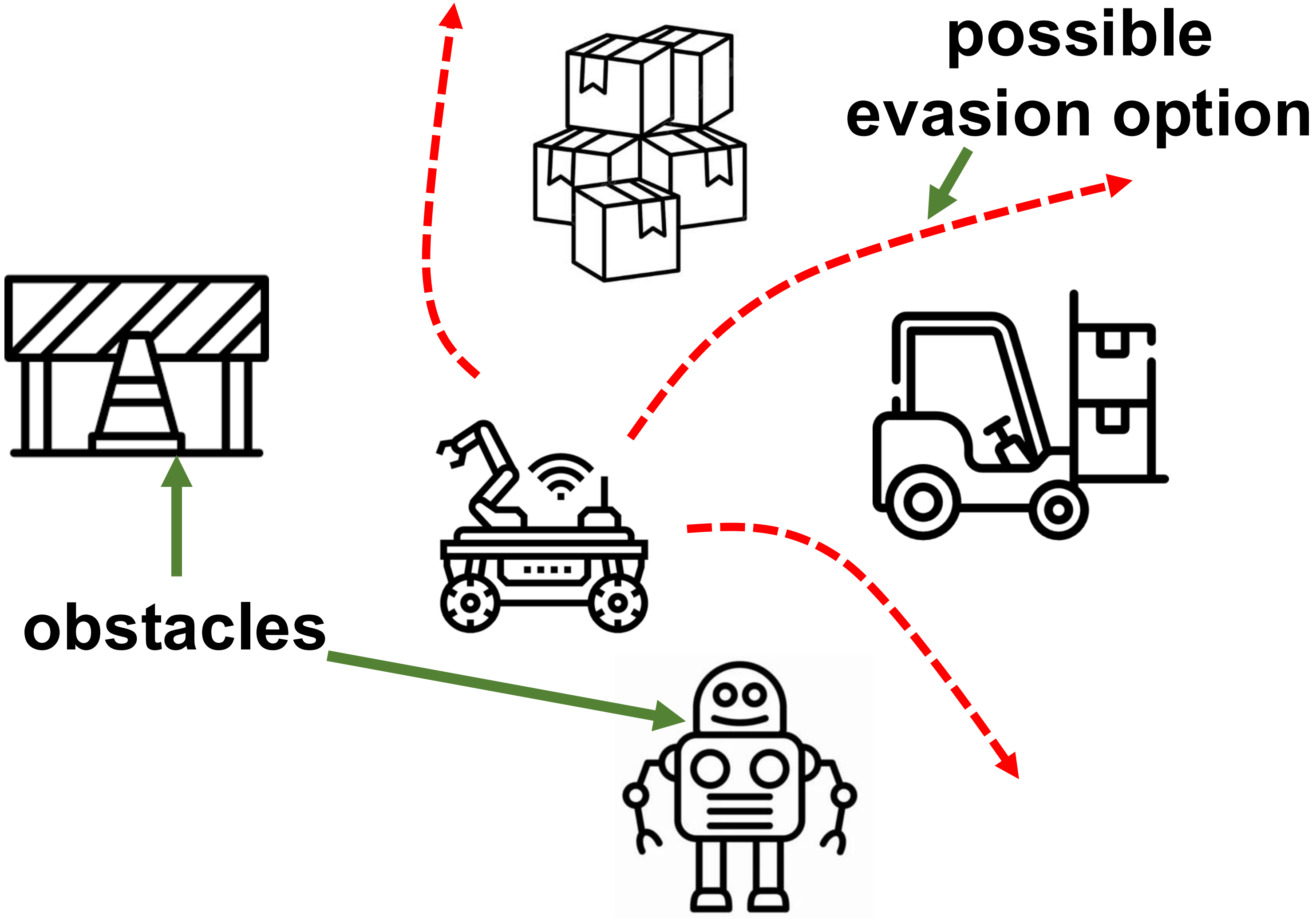
Criticality Evasion

Uncertainty adds to these challenges

Evaluate *safety* of each possible evasion option

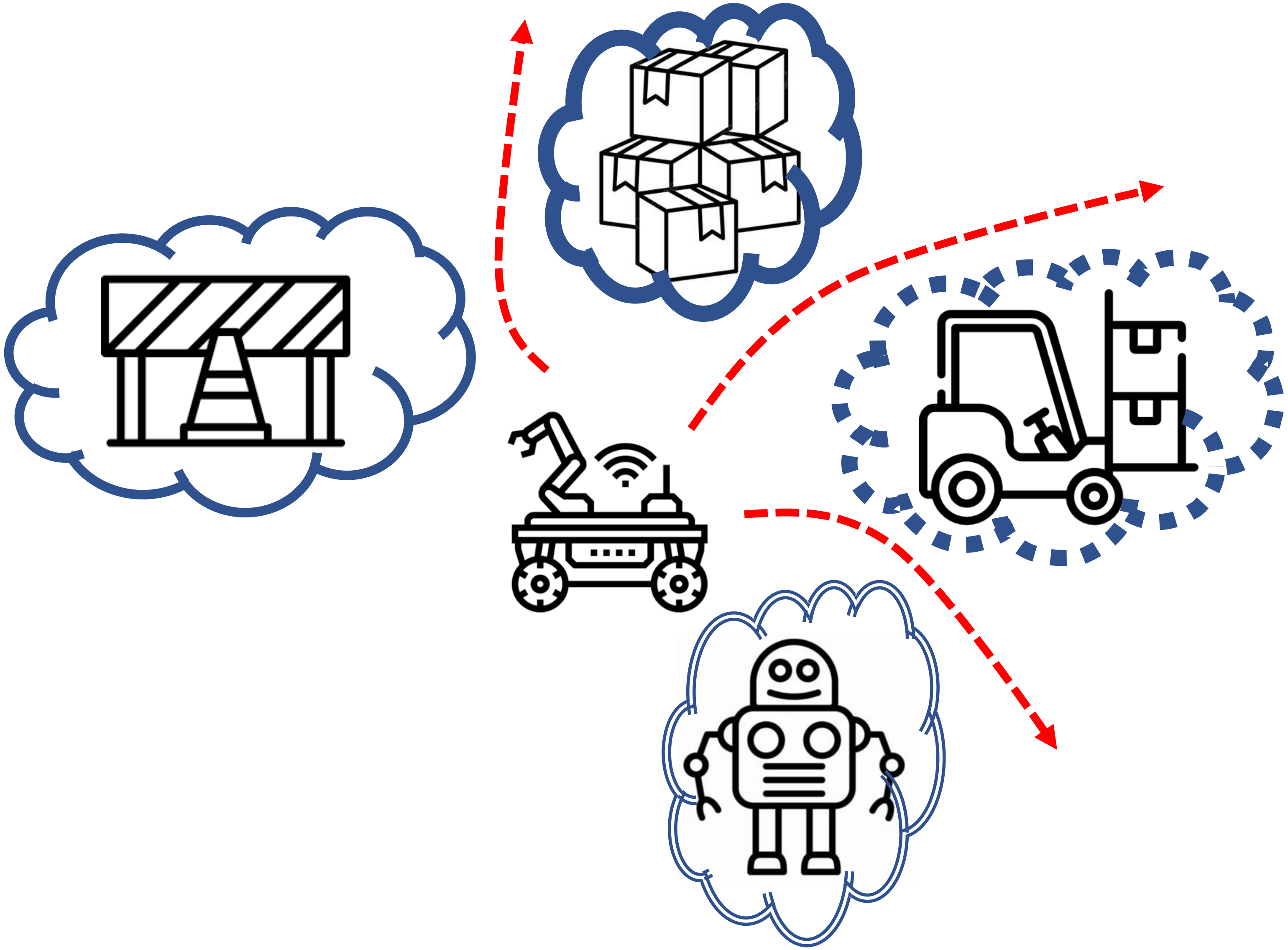
Pursue the *safest* option/path

Must be *fast* for it to be online



Criticality Evasion *Under Uncertainties*

Uncertainty adds to these challenges



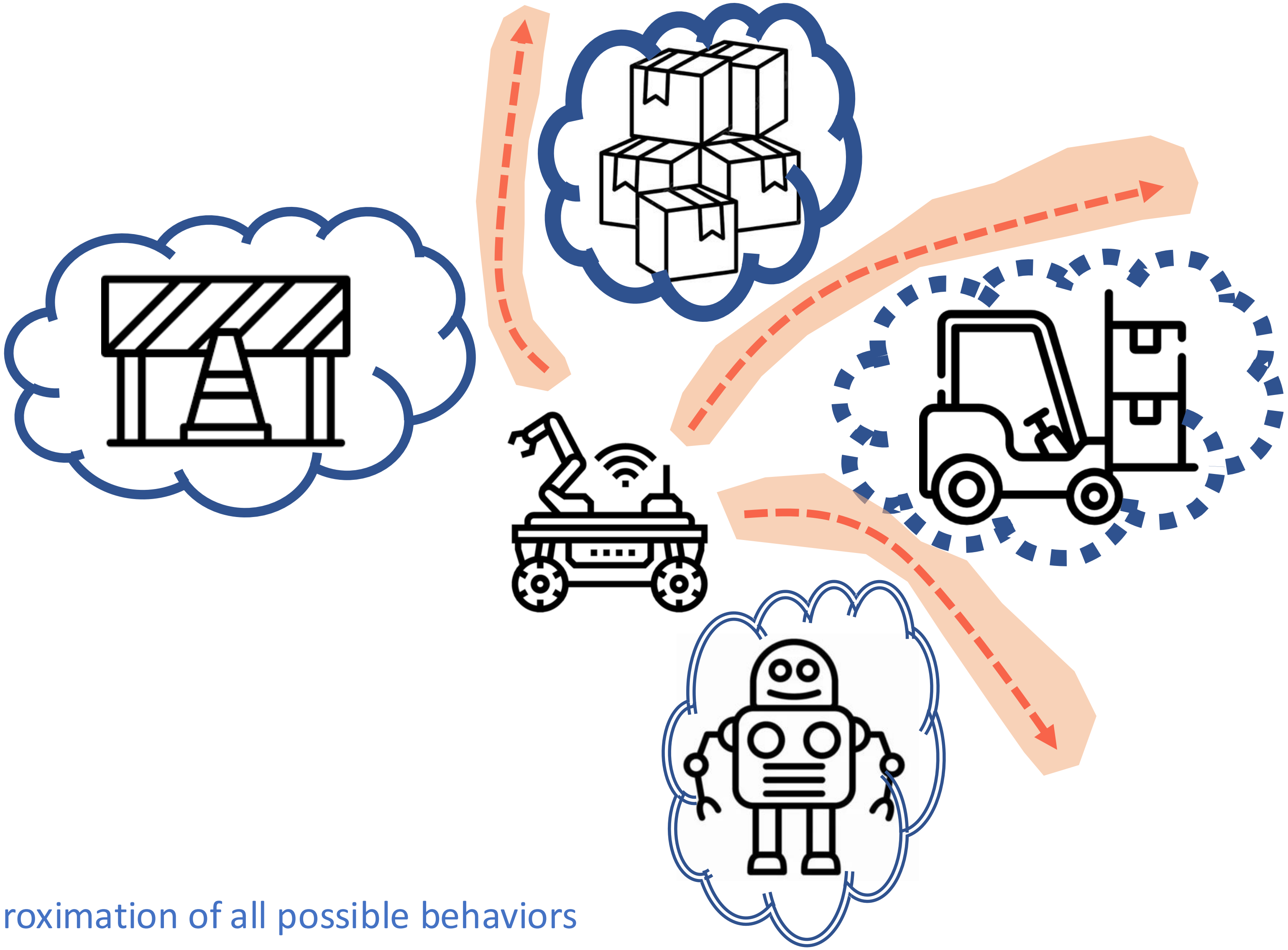
Evaluate *safety* of each possible evasion option

Pursue the *safest* option/path

Must be *fast* for it to be online

Traditional Methods of Criticality Evasion

Using *reachable sets* to analyze safety of each option

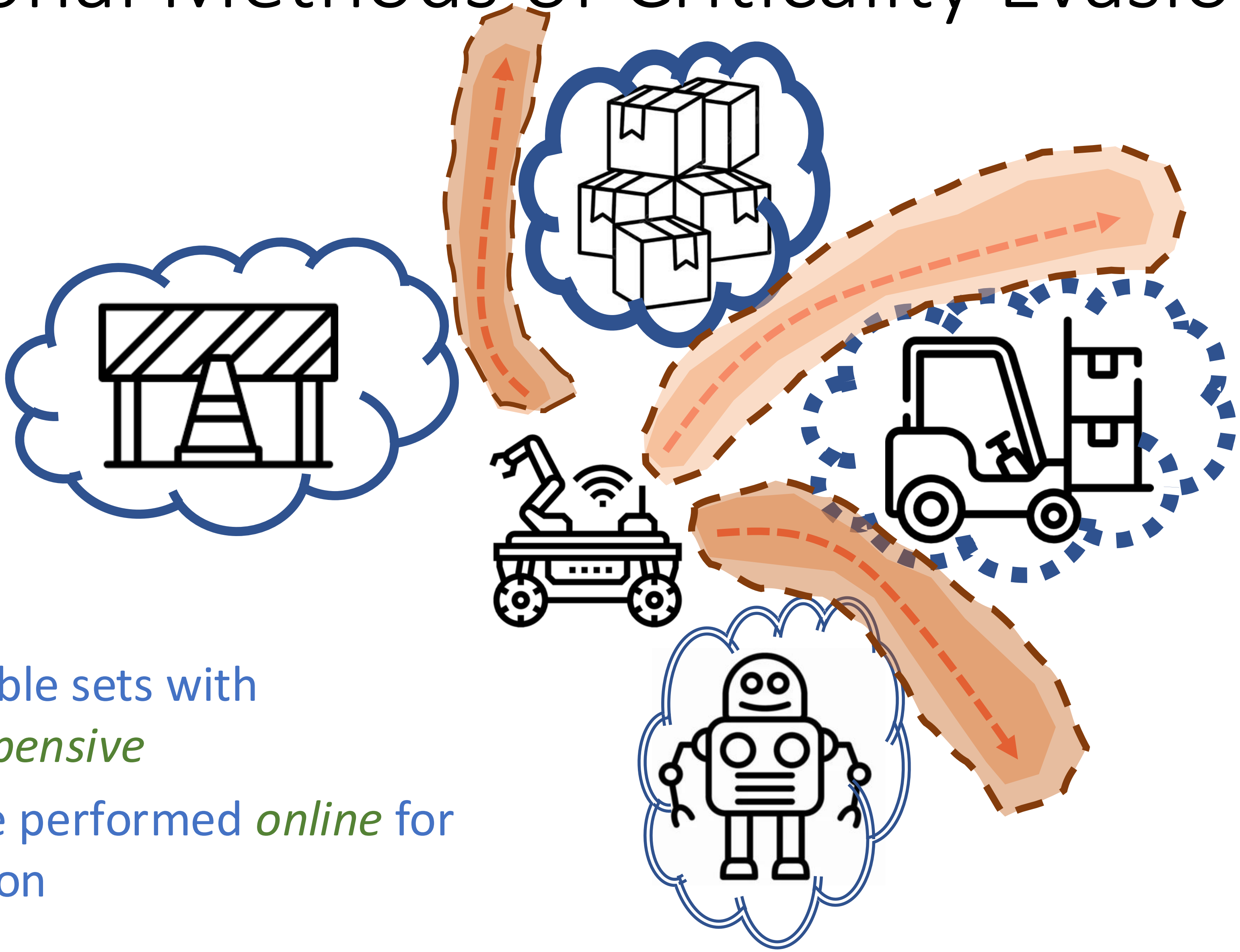


Reachable Sets: Over approximation of all possible behaviors

Traditional Methods of Criticality Evasion

Using *reachable sets* to analyze safety of each option

Uncertainties have its own impact

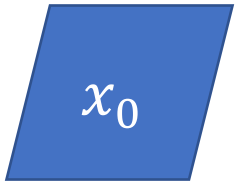


Computing reachable sets with uncertainties is *expensive*

Infeasible to be performed *online* for criticality evasion

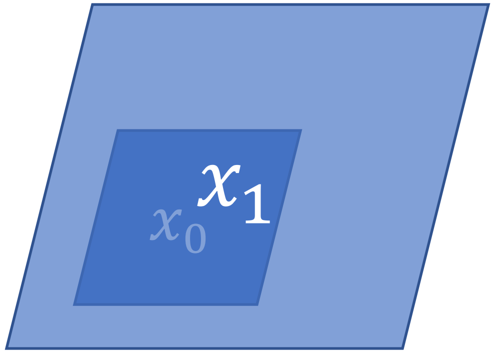
Reachable Sets

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x}$$



Reachable Sets

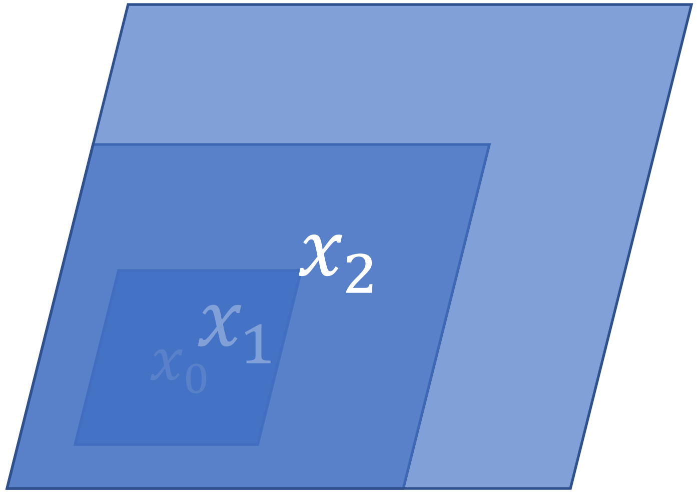
$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x}$$



$$x_1 = e^{A\times 1} \cdot x_0$$

Reachable Sets

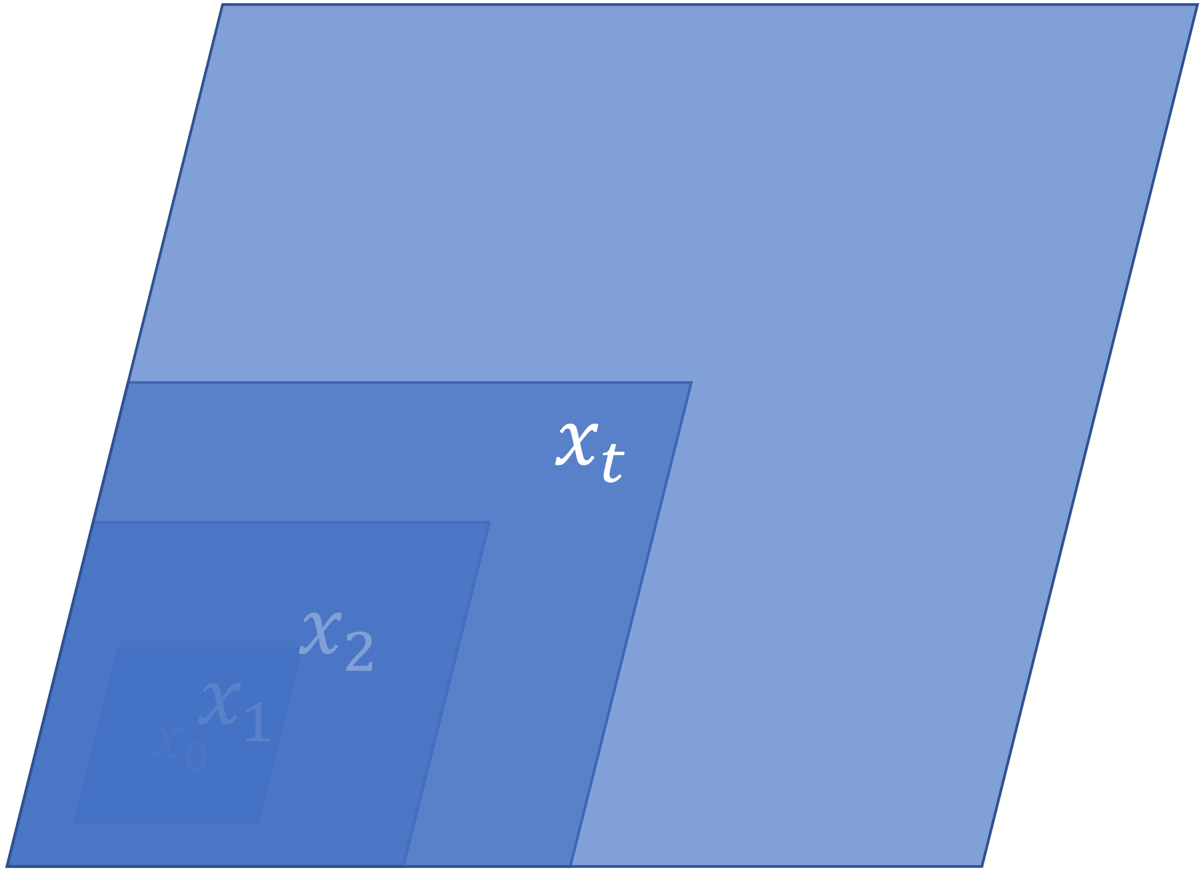
$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x}$$



$$x_2 = e^{A\times 2} \cdot x_0$$

Reachable Sets

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x}$$



$$x_t = e^{A \times t} \cdot x_0$$

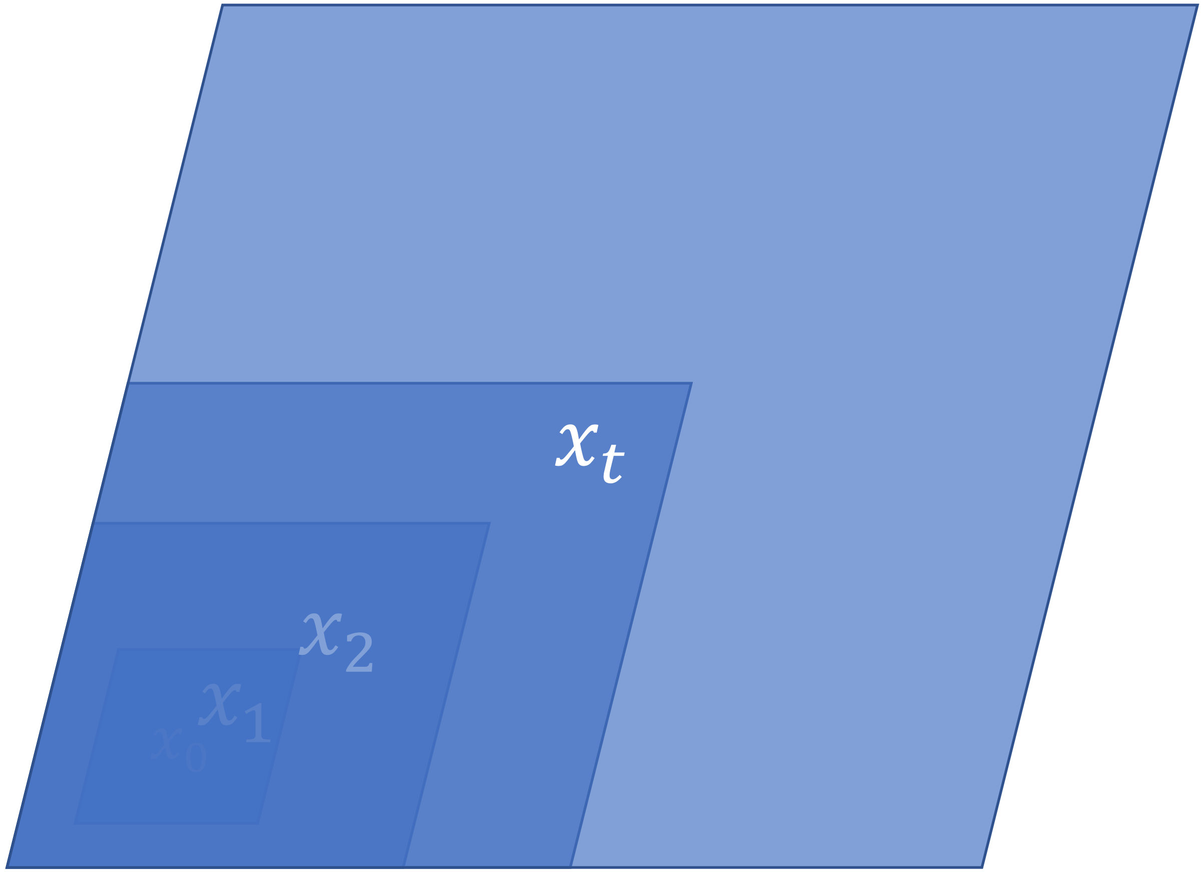
Reachable Sets

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x}$$

What if now A has uncertainties?

I.e., $\dot{x} = (A + \Lambda)\,x$, where Λ is an interval matrix

Computing $e^{(A+\Lambda)t}$ is computationally expensive (if not impossible)

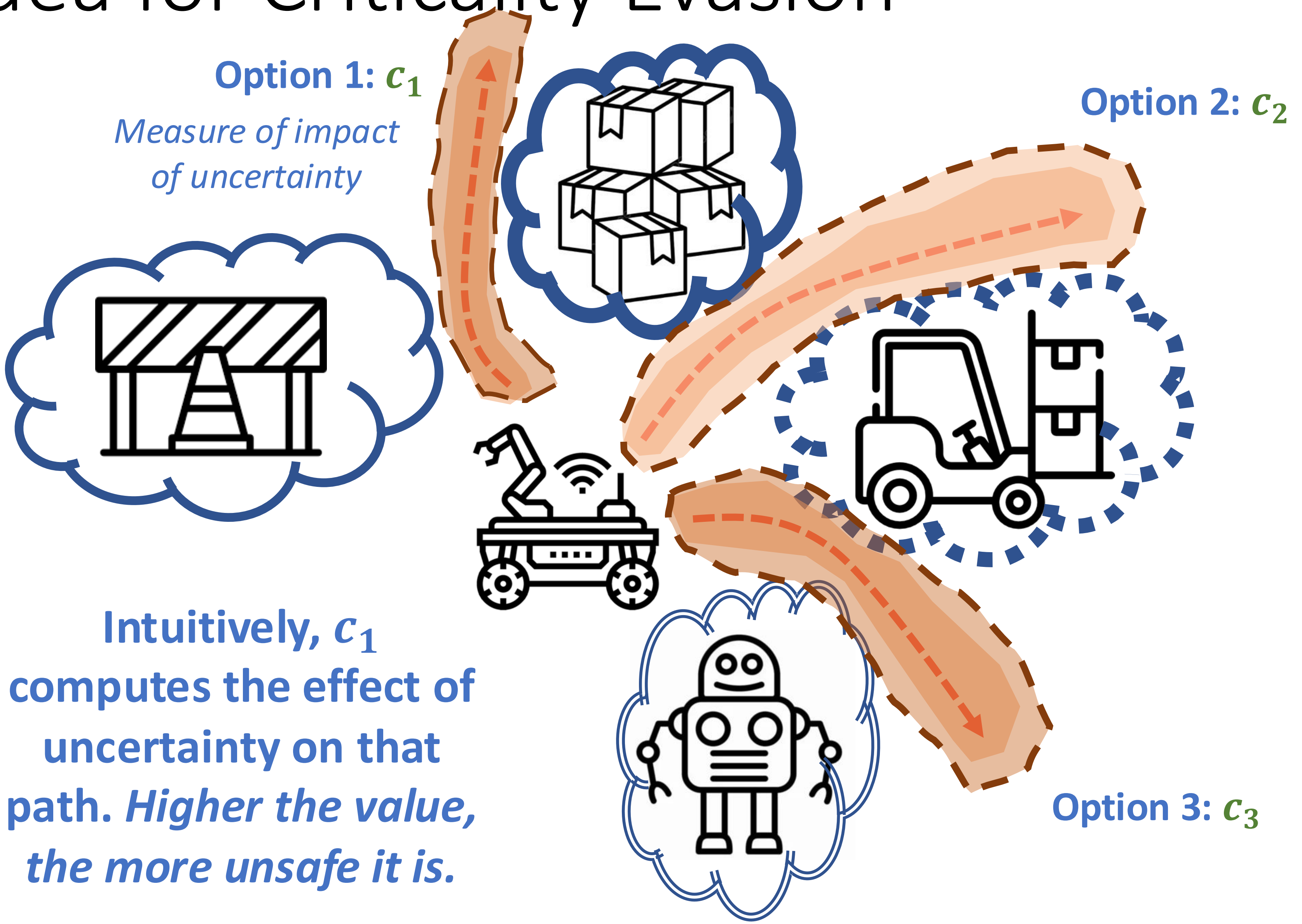


$$x_t = e^{A\times t} \cdot x_0$$

Main Idea for Criticality Evasion

Main Idea:

- *Quantify* the effect of uncertainty *without computing reachable sets*
- Using Perturbation Theory

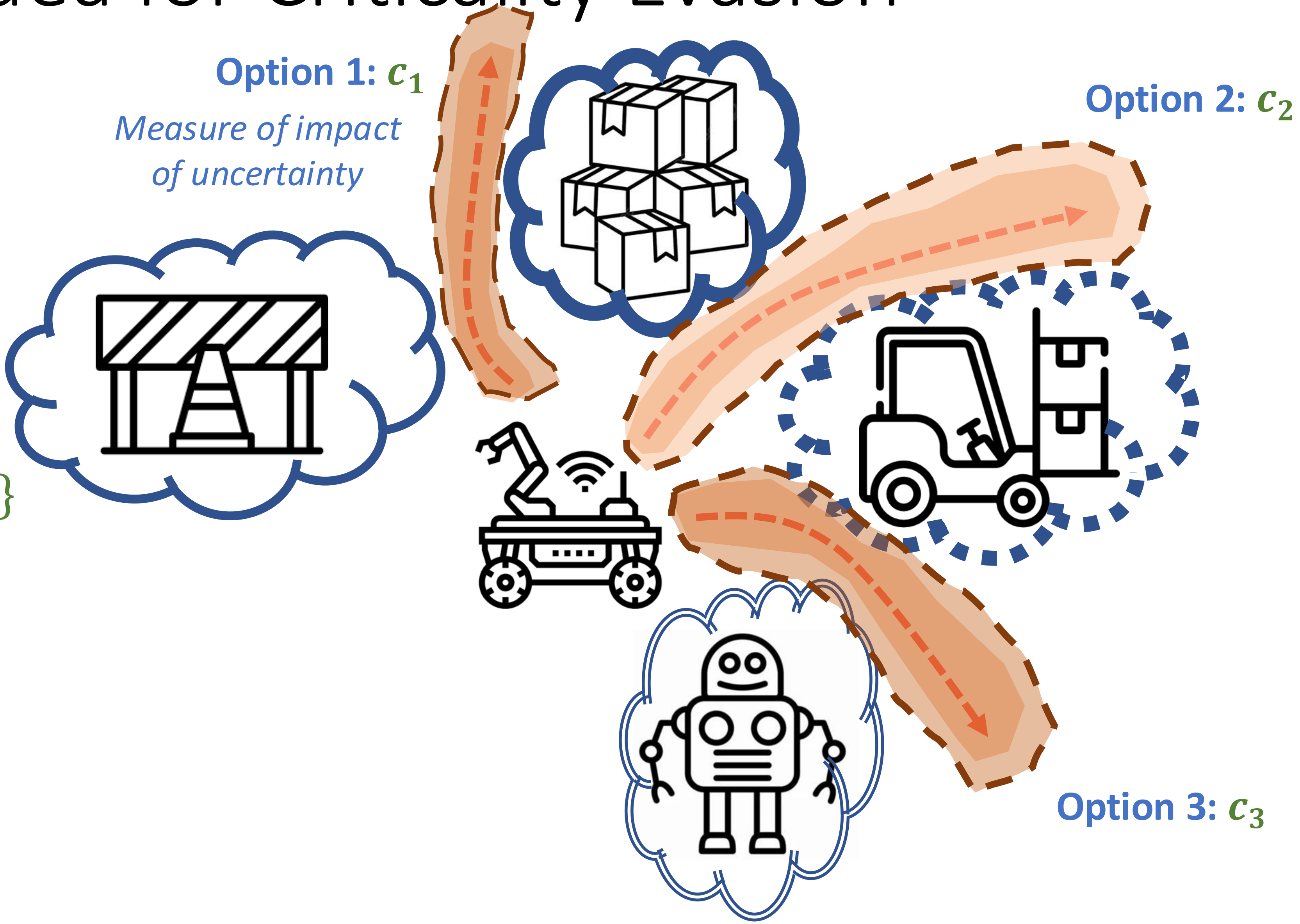


Intuitively, c_1 computes the effect of uncertainty on that path. *Higher the value, the more unsafe it is.*

Main Idea for Criticality Evasion

Pursue the option with minimum impact

Pursue the option with $\min\{c_1, c_2, c_3\}$



Modeling Options with Uncertainties

Model $Option_i$ as:

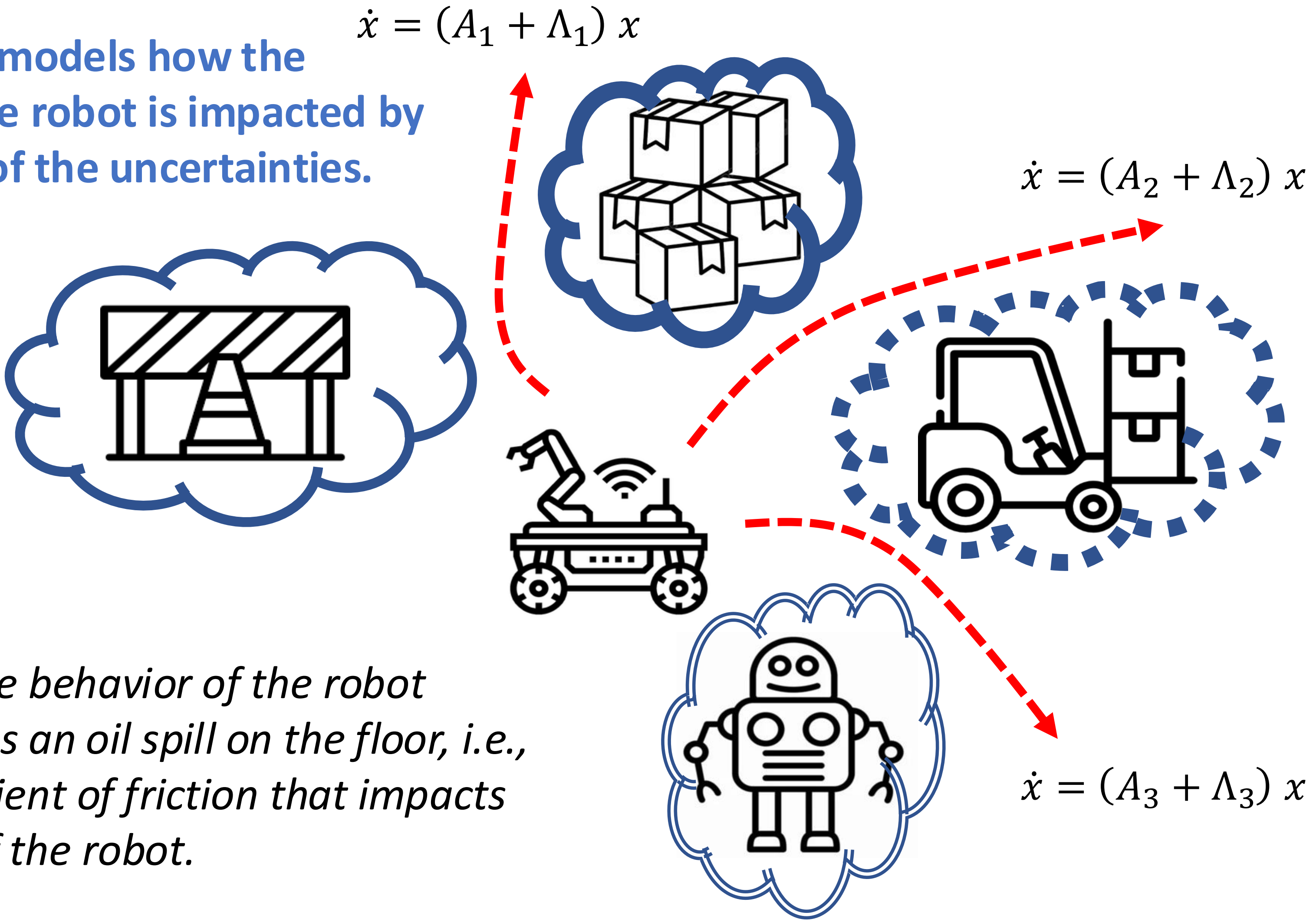
$$\dot{x} = (A_i + \Lambda_\mathrm{i})\, x$$

System states

Impact of uncertainties

Nominal behavior (without uncertainties) of the robot pertaining to $option_i$

Intuitively, Λ_i models how the behavior of the robot is impacted by the presence of the uncertainties.



Example: How the behavior of the robot changes if there is an oil spill on the floor, i.e., change in coefficient of friction that impacts the navigation of the robot.

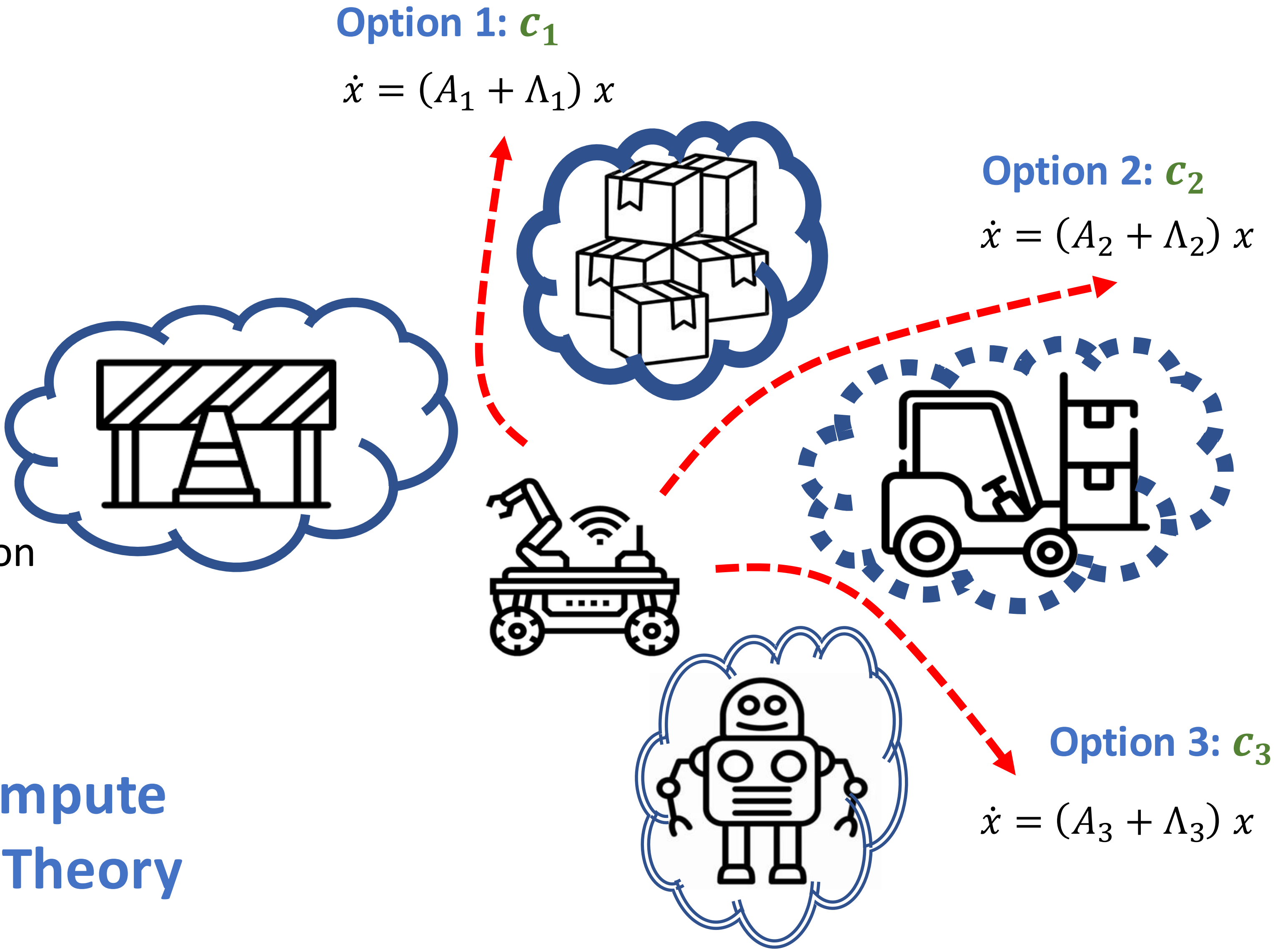
Compute Criticality

$$\dot{x} = (A_i + \Lambda_\mathrm{i})\ x$$

Compute Criticality

$$\phi_{(A_i,\Lambda_i)}(t) = c_i$$

Bounded time-horizon



We propose 3 methods to compute $\phi_{(A,\Lambda)}(t)$ using Perturbation Theory

Method I: Compute $\phi_{(A,\Lambda)}(t)$

- Using p-approximation
  - $p_{n-1}(x) = \sum_{k=0}^{n-1} \frac{x^k}{k!}$
- Derive an upper bound of $\phi_{(A,\Lambda)}(t)$
  - Using p-approximation of $||A||_2$
- Intuitively,
  - $\phi_{(A,\Lambda)}(t) = \boxed{\text{Compute } p\text{-approximation of 2-norm of } A} \times \boxed{\text{Exponent of a combination of } p\text{-approximation of } \Lambda \text{ \& A and time } t}$
- Referred as **Kagstrom1**

Results extended from B. Kagstrom, “Bounds and perturbation bounds for the matrix exponential,” BIT Numerical Mathematics, vol. 17, no. 1, pp. 39–57, 1977

Method II: Compute $\phi_{(A,\Lambda)}(t)$

- Using Condition Number
  - Condition number of matrix M: $K(M) = ||M|| \cdot ||M^{-1}||$
- Jordan Decomposition of A
  - SJS^{-1}, and D is diagonal matrix, *s.t.*,
    - $||D^{-1}JD|| \leq \epsilon$
- Specifically,
  - $\phi_{(A,\Lambda)}(t) =$ **Compute condition number of A** $\times$ **Exponent of a combination of condition numbers of Λ & A and time t**
- Referred as **Kagstrom2**

Results extended from B. Kagstrom, "Bounds and perturbation bounds for the matrix exponential," BIT Numerical Mathematics, vol. 17, no. 1, pp. 39–57, 1977

Method III: Compute $\phi_{(A,\Lambda)}(t)$

- Using Eigen Values
  - $\alpha(M)$ is the spectral abscissa of matrix M

- Specifically,
  - $\phi_{(A,\Lambda)}(t) = \boxed{\text{t times 2-norm of Λ}} \times \boxed{\text{Exponent of a combination of (condition number of A, 2-norm of Λ) and time t}}$

- Referred as **Loan**

Results extended from C. V. Loan, "The sensitivity of the matrix exponential," SIAM Journal on Numerical Analysis, vol. 14, no. 6, pp. 971–981, 1977.

Comparison: Methods I, II, & III

Open Question: The best-performing method for a given system remains unknown — must be identified empirically.

For all the three methods, computing 2-norm of Λ is required

Method I

Apply it for systems where computing p-approximation and 2-norm of A & Λ is efficient

Method II

Apply it for systems where computing condition number and Jordan Decomposition of A is efficient

Method III

Apply it for systems where computing spectral radius of A is efficient

Overall Approach

- For each $Option_i$:
  - Compute criticality $c_i = \phi_{(A_i,\Lambda_i)}(t)$, which quantifies the impact of the uncertainty
    - Using the tightest of the three available upper bounds (Kagstrom1, Kagstrom2, Loan)

- Once we have computed all the criticality factors $\{c_1, c_2, \cdots, c_n\}$, we sort the navigation choices in increasing order of these values.

- We then verify the safety of the navigation choices in this order using reachability analysis

- Pursue the first encountered safest option

Experiments

- **Benchmarks.** Linear dynamics models from ARCH workshop suite
  - Dimensions: 2–10; Perturbations: 2–4 entries in system matrix; Time step: Up to 20
- All three methods are extremely efficient, even for high-dimensional systems (10).
  - Took less than 0.5 seconds.
- Loan performed the best in all cases, with a few a joint winners
- **Stress-test (t = 50–70):** Kagstrom1/2 bounds grew **exponentially** (≈50×→500×), while **Loan** grew **near-linearly** (≈25×→40×), yielding significantly **tighter bounds at higher times**.
- This efficiency makes them particularly suitable for real-time decision-making on prioritizing navigation choices in uncertain environments

Conclusion & Future Work

Thank You

- Safety-critical navigation requires rapid decisions, but traditional verification is often slow.
- We proposed an efficient **ranking method** that prioritizes navigation choices by sensitivity to uncertainty.
- Our method achieves **sub-0.5s performance** even on large-scale benchmarks.
- *Future work:* extend to **reachable set computation** and **structure-aware reachability** (that goes beyond norms of A and Λ).