Algorithmic Human-Robot Interaction

Trajectory Optimization

&

Project Ideas

Today: Trajectory Optimization & Project Brainstorming

Making Robots Move (how we want them to)

- Optimizing motion plans

Research Projects

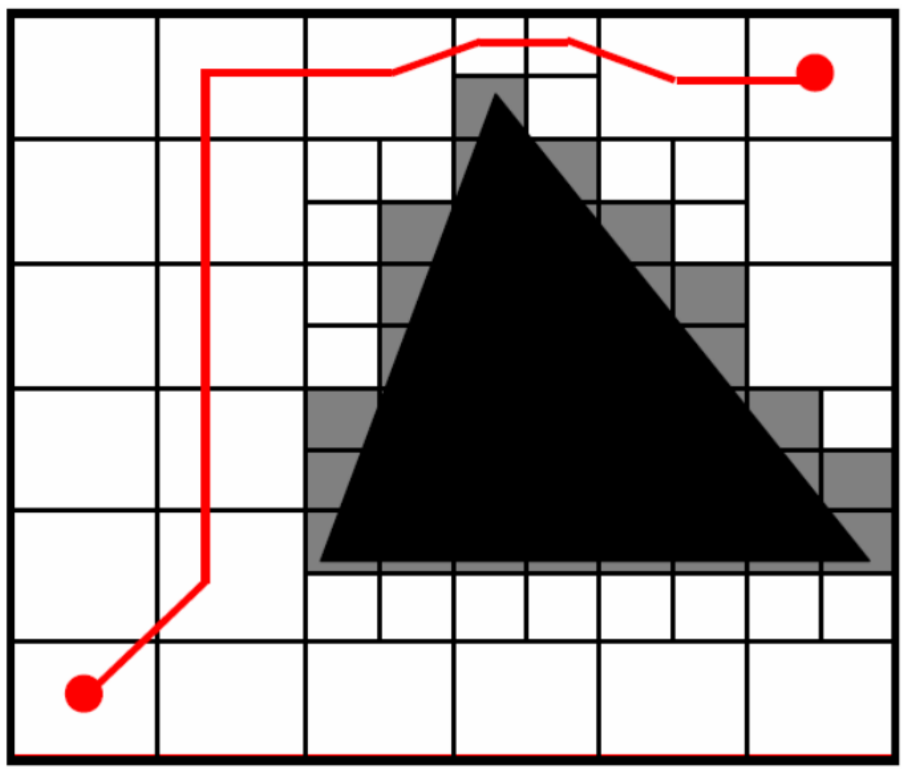
- Broad overview of open HRI problems not covered last time
- Rapid-fire Project Discussions

Last Time…

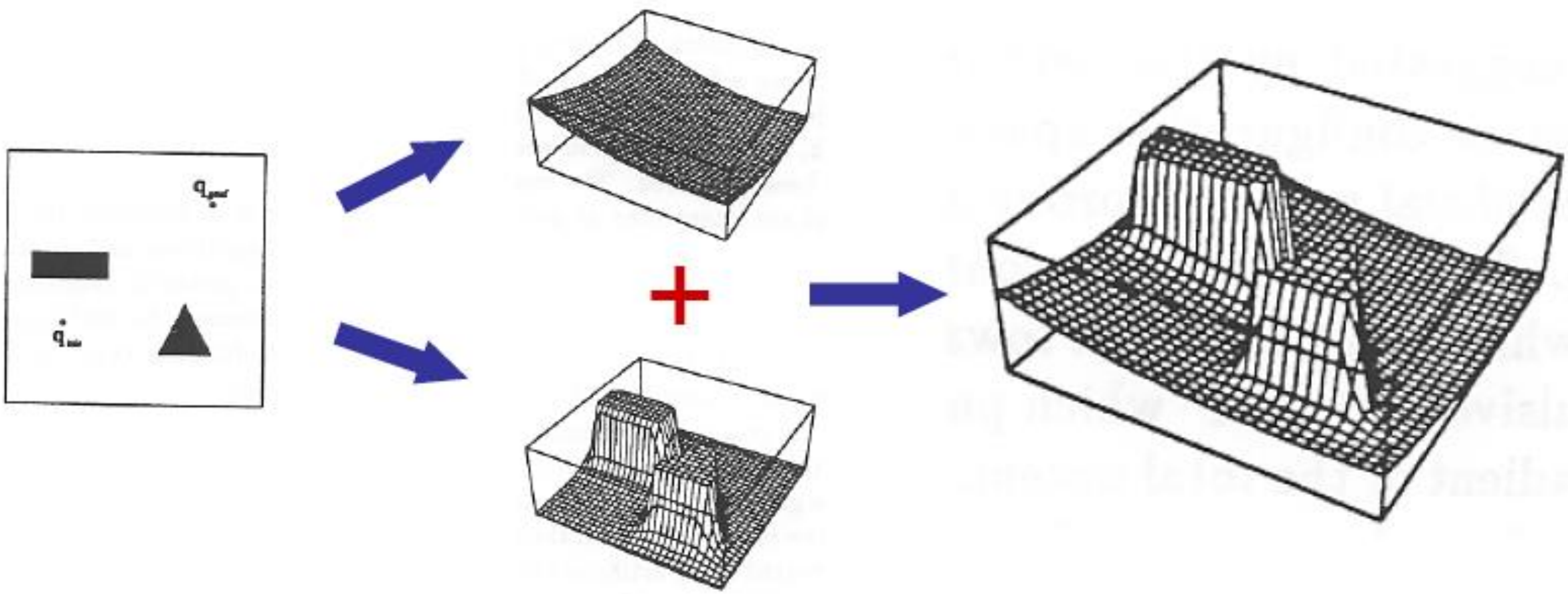
Cell Decomposition

Not necessarily *complete*

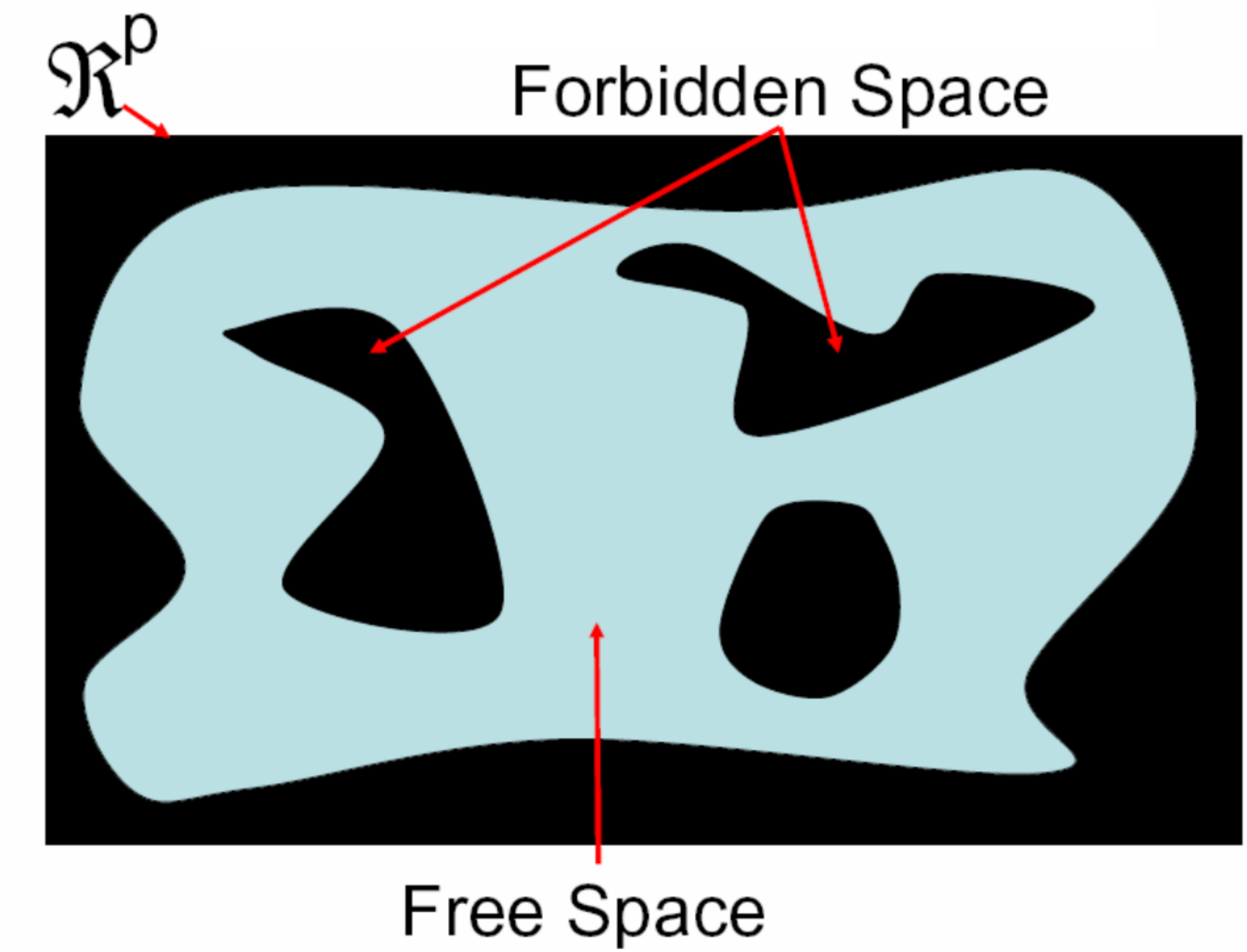
(Complete: If a solution exists, it will eventually be found)



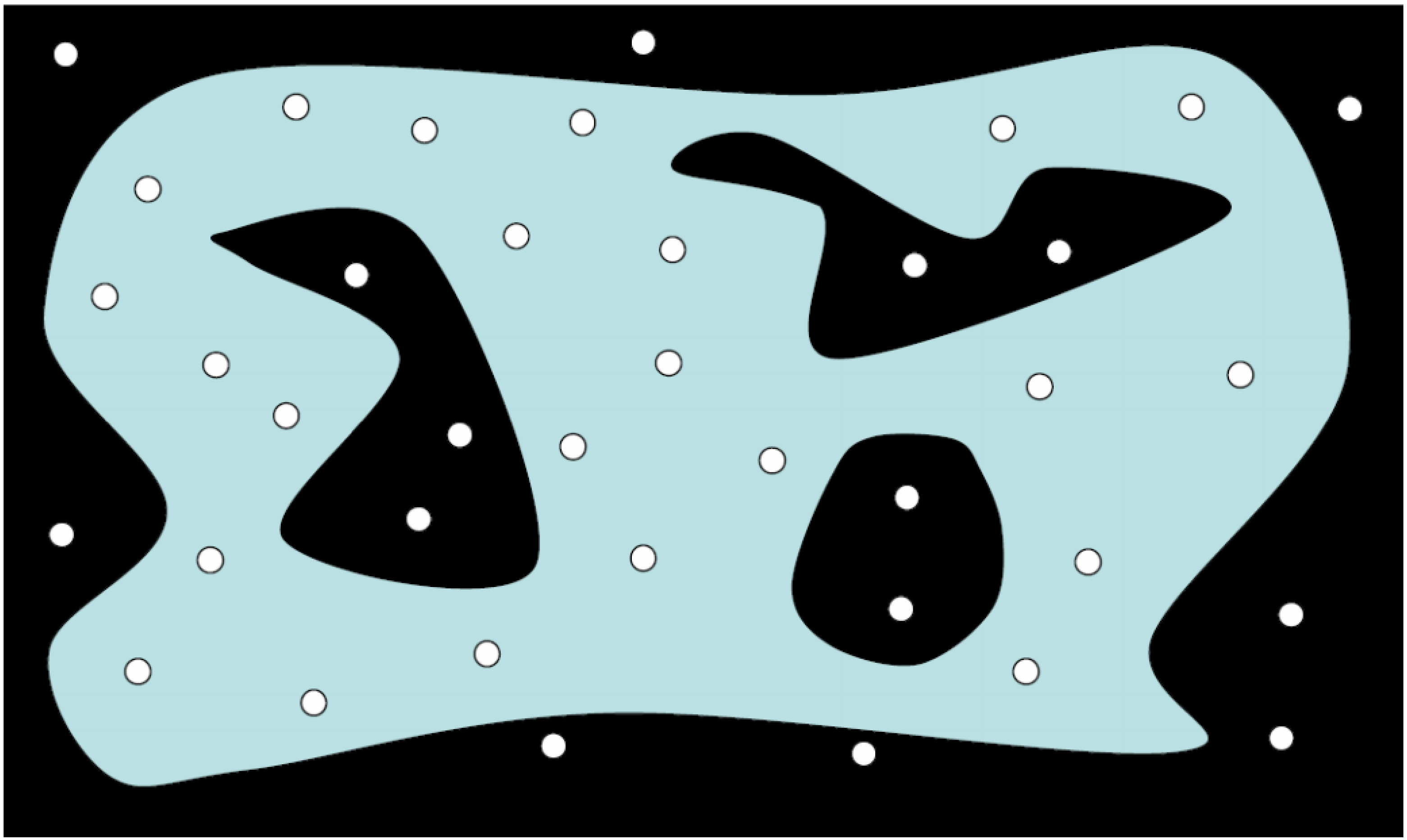
How Does It Work?



Probabilistic Road Map (PRM)

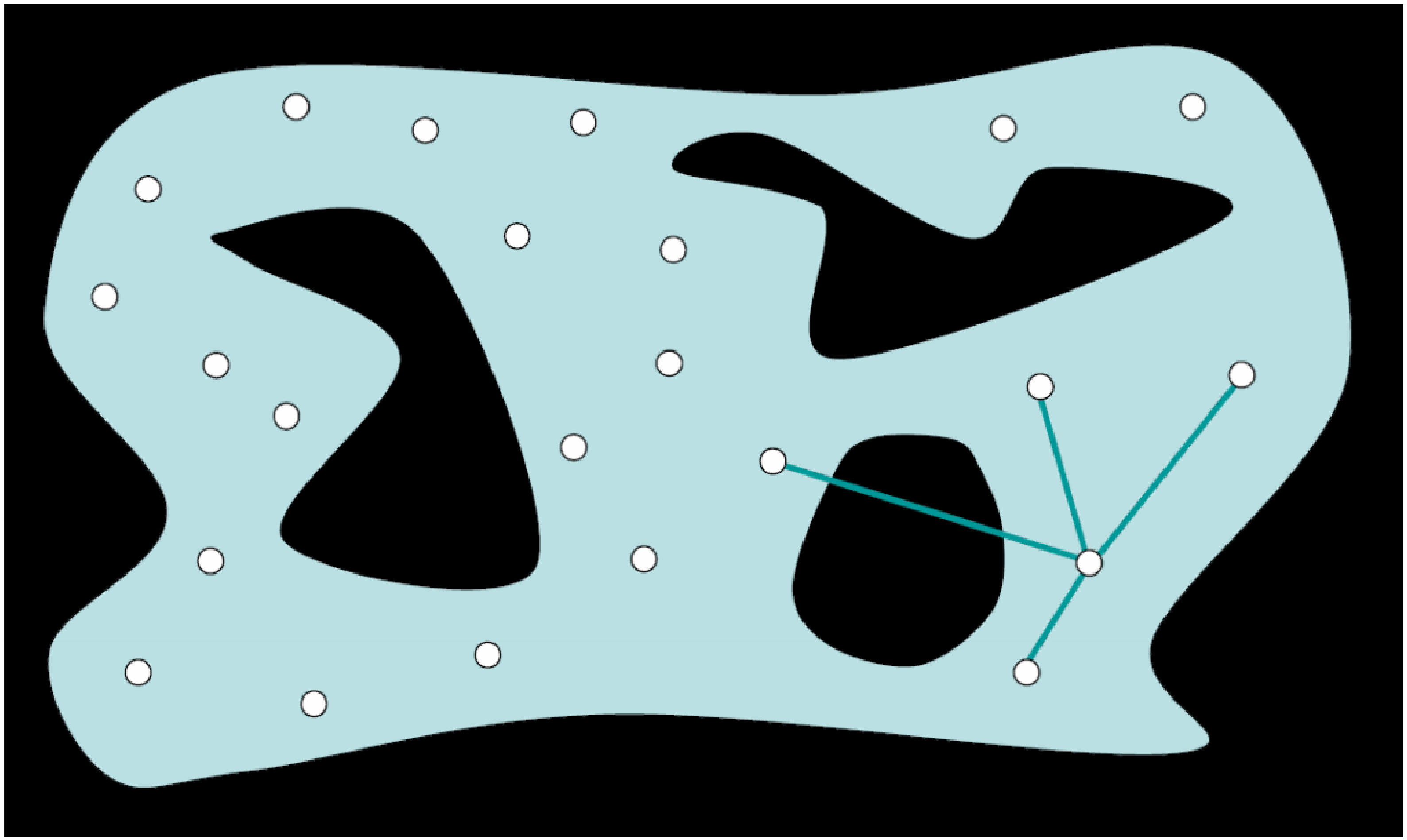


Probabilistic Road Map (PRM)



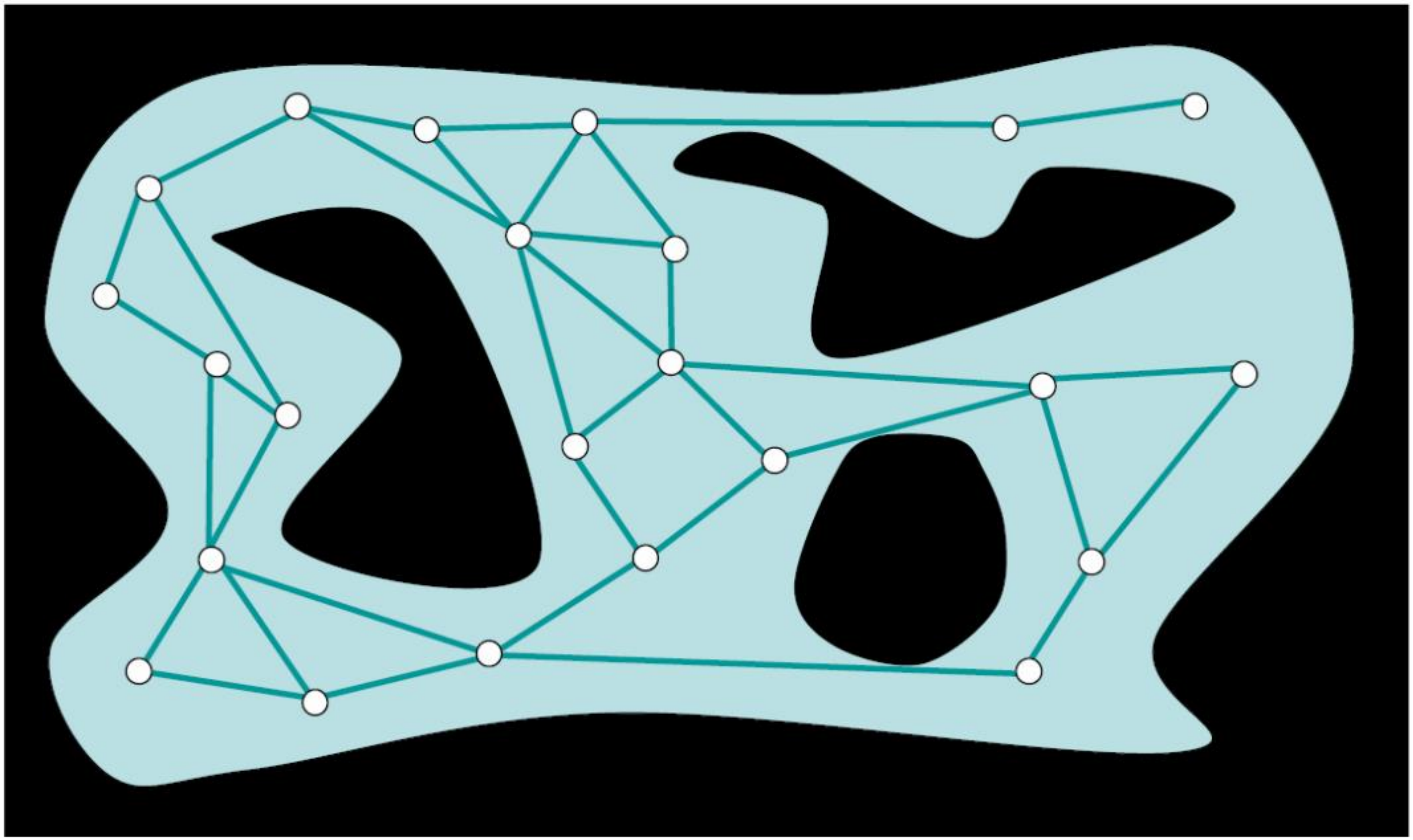
Sample random locations!

Probabilistic Road Map (PRM)



Remove points in forbidden areas
Link each point to its K nearest neighbors

Probabilistic Road Map (PRM)

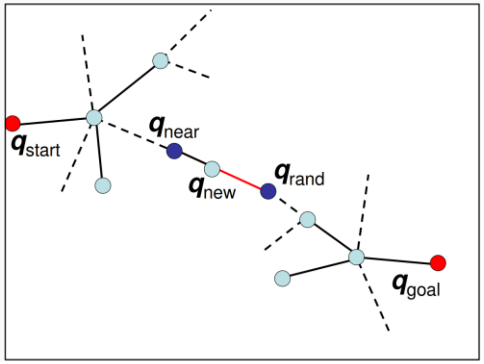


Remove edges crossing forbidden areas

Rapidly-exploring Random Trees

```
Algorithm BuildRRT
  Input: Initial configuration q_init, number of vertices in RRT K, incremental distance Δq)
  Output: RRT graph G

  G.init(q_init)
  for k = 1 to K
    q_rand ← RAND_CONF()
    q_near ← NEAREST_VERTEX(q_rand, G)
    q_new ← NEW_CONF(q_near, q_rand, Δq)
    G.add_vertex(q_new)
    G.add_edge(q_near, q_new)
  return G
```



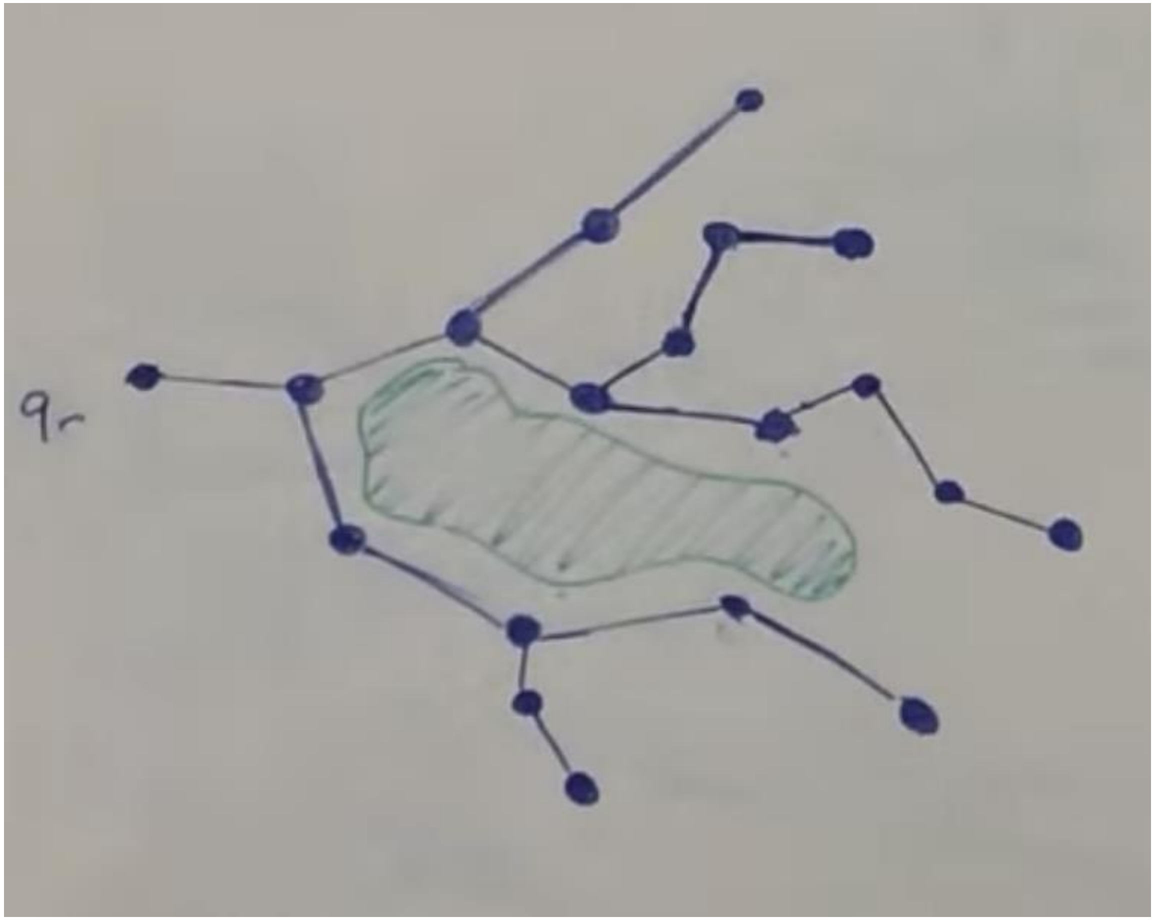
RRT: Limitations

- RRT fails to converge to optimal solutions
  - Early solutions end up constraining the search
- RRT* guarantees asymptotic optimality (convergence to optimal solutions)
- RRT and RRT* require the same (asymptotic) computational resources

RRT

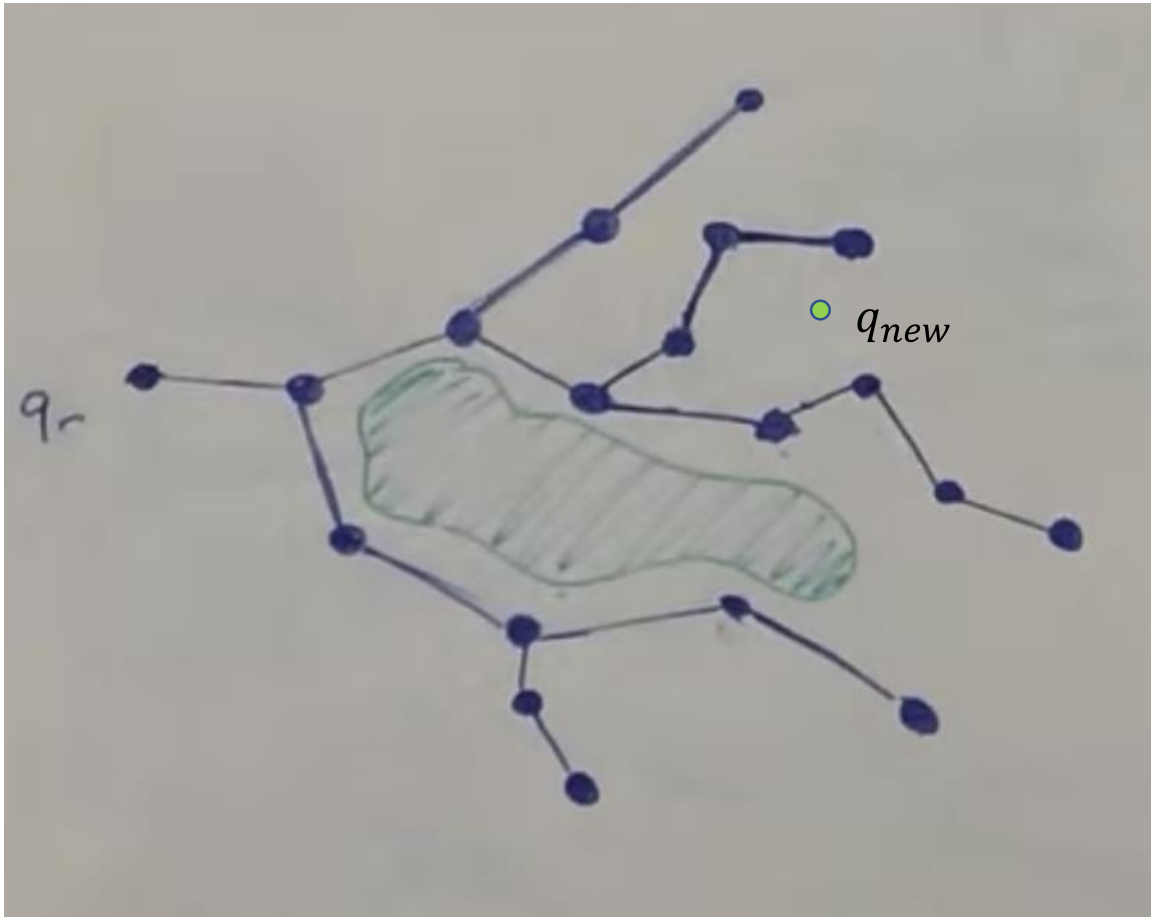
RRT*

RRT*



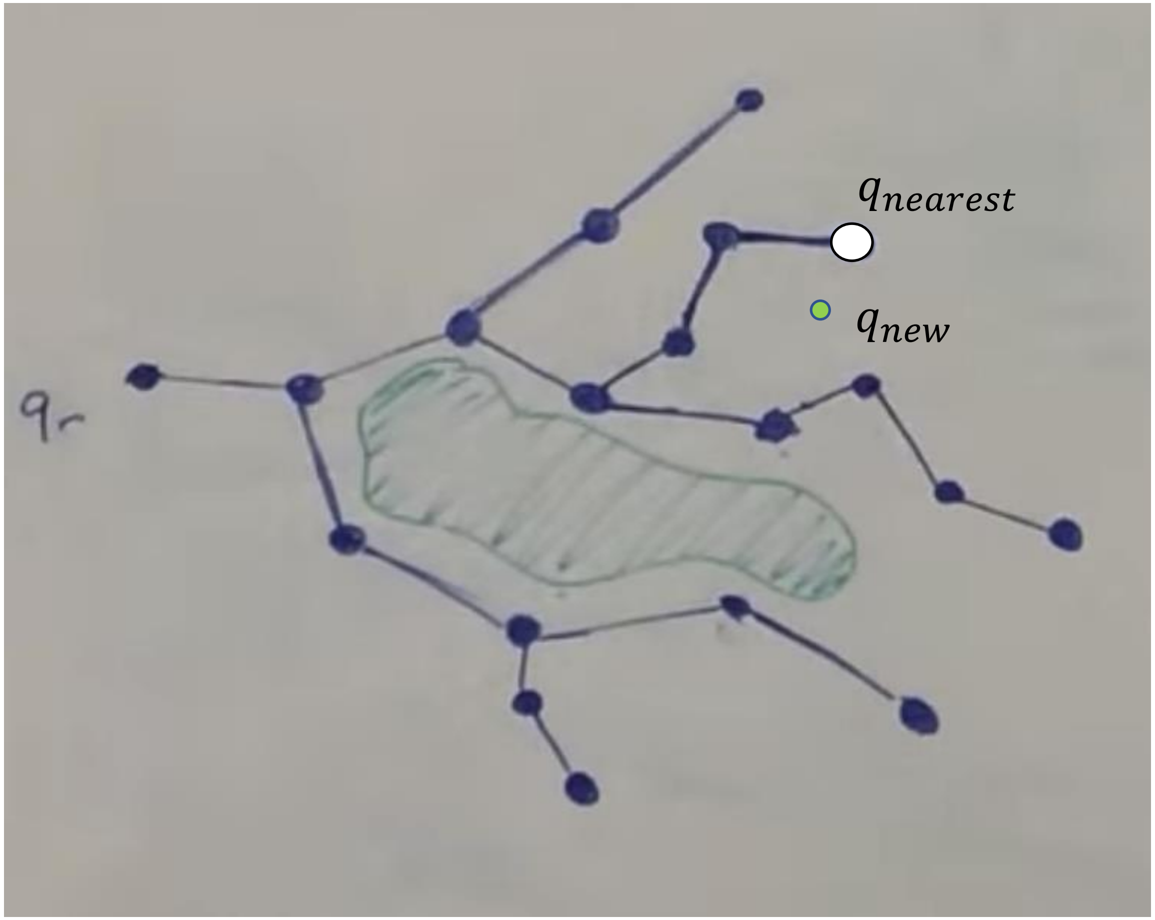
https://www.youtube.com/watch?v=JM7kmWE8Gtc

RRT*: Sample



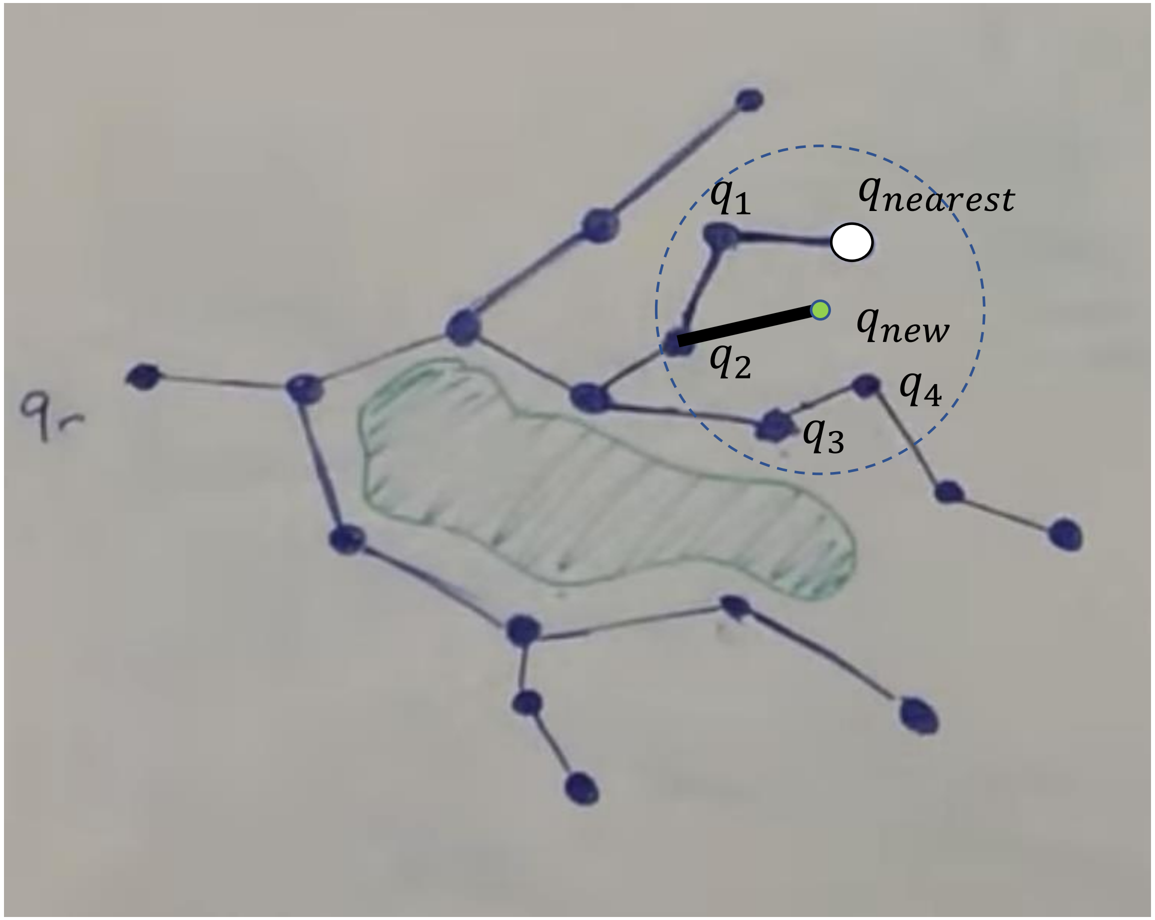
https://www.youtube.com/watch?v=JM7kmWE8Gtc

RRT*



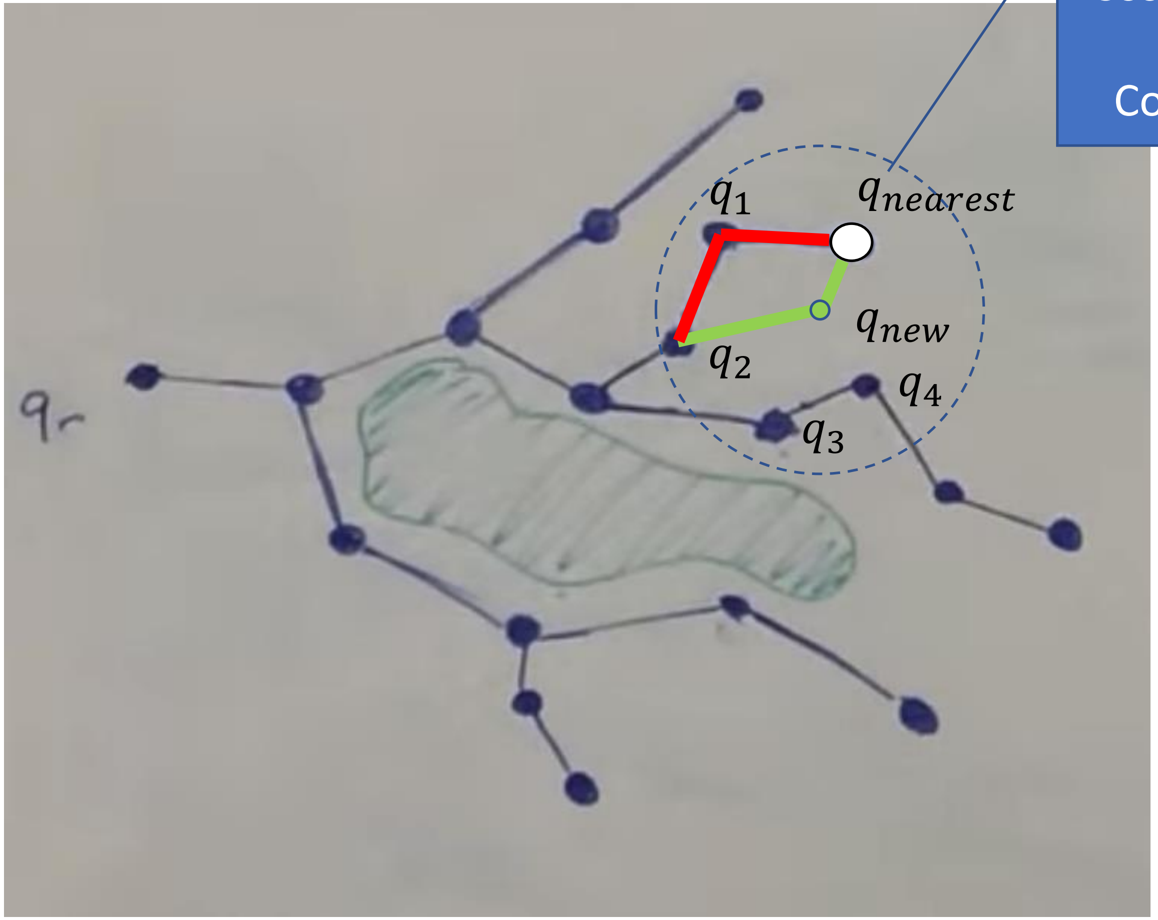
https://www.youtube.com/watch?v=JM7kmWE8Gtc

RRT*: Connect cheapest path



https://www.youtube.com/watch?v=JM7kmWE8Gtc

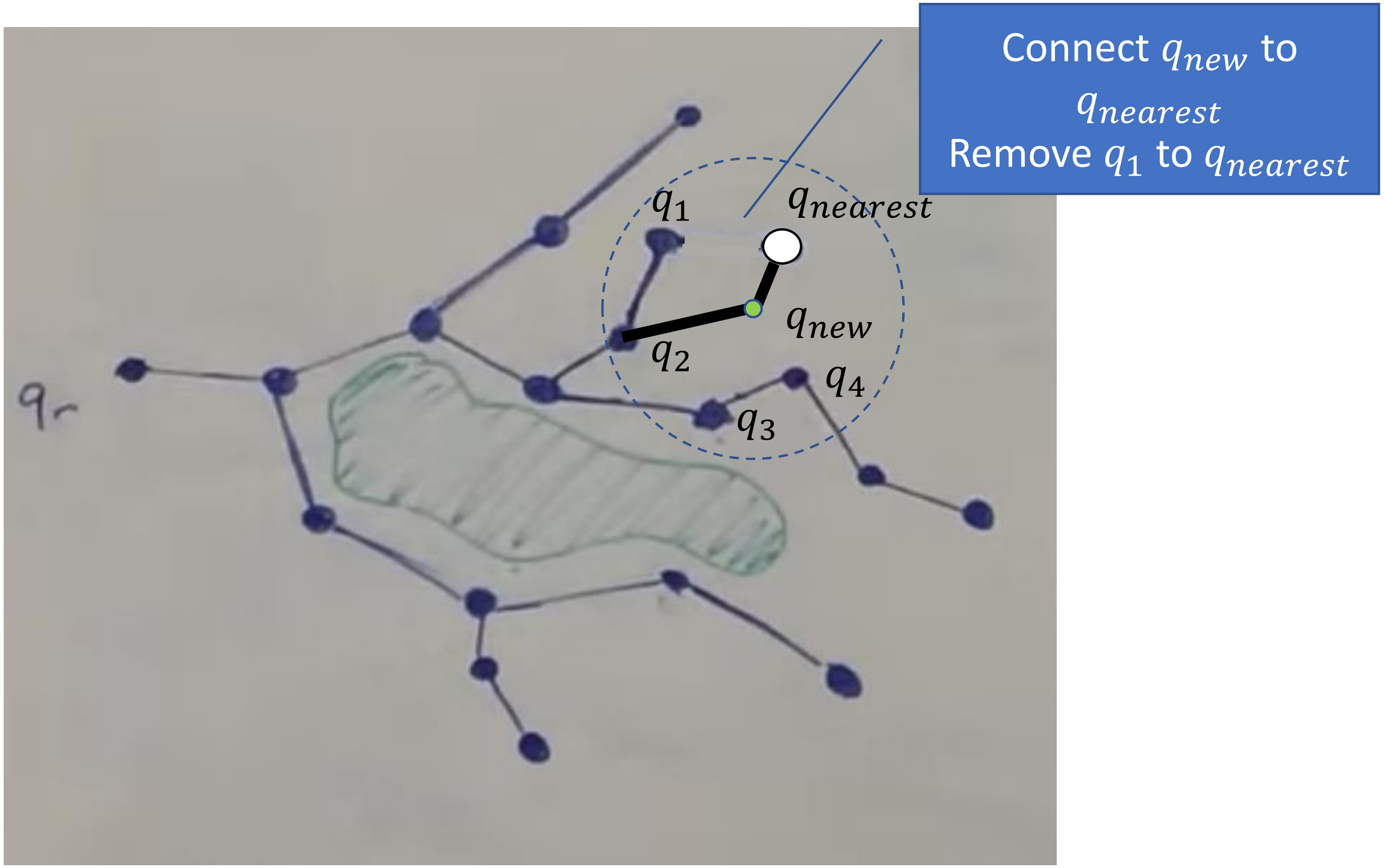
RRT*: Re-wire Graph



Cost of green route
<
Cost of red route

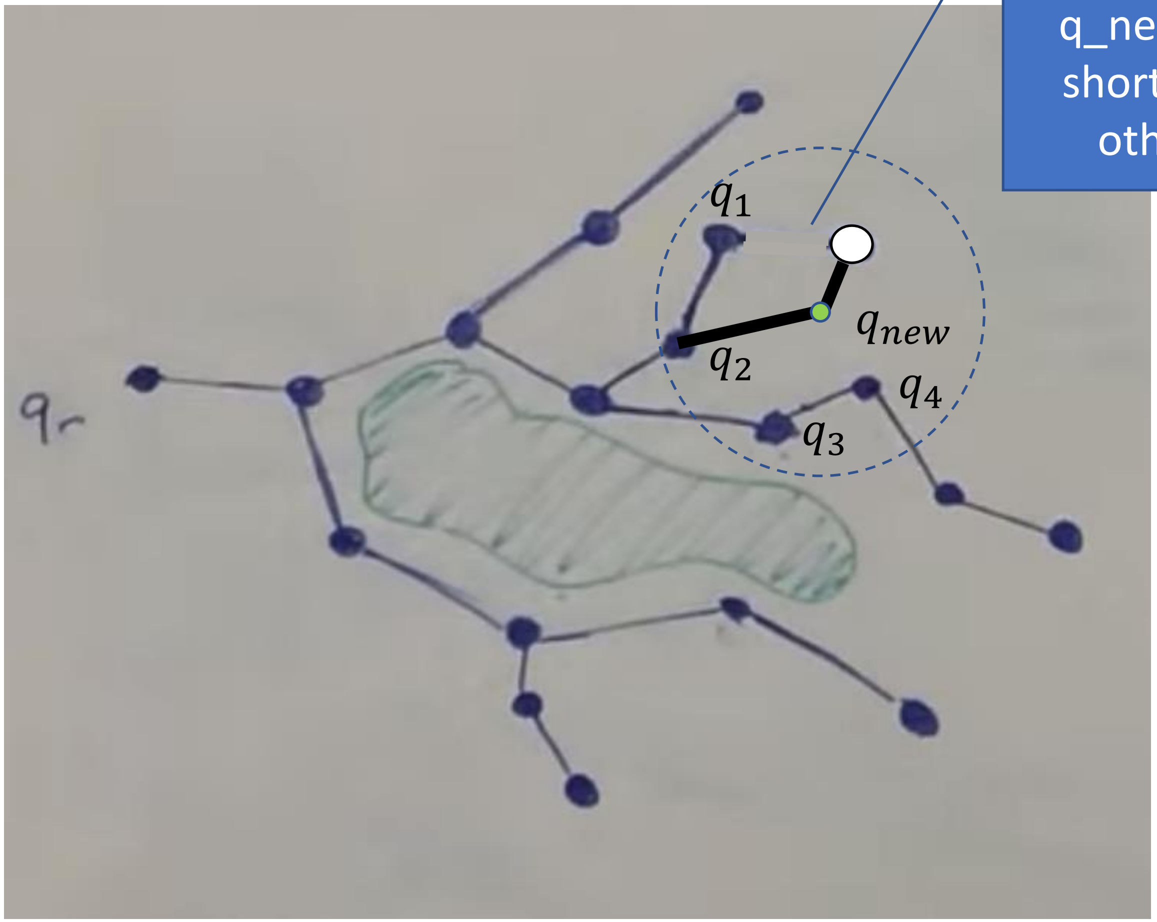
https://www.youtube.com/watch?v=JM7kmWE8Gtc

RRT*: Re-wire Graph



https://www.youtube.com/watch?v=JM7kmWE8Gtc

RRT*: Re-wire Graph



Compare path through q_new with existing shortest path for all other neighbors

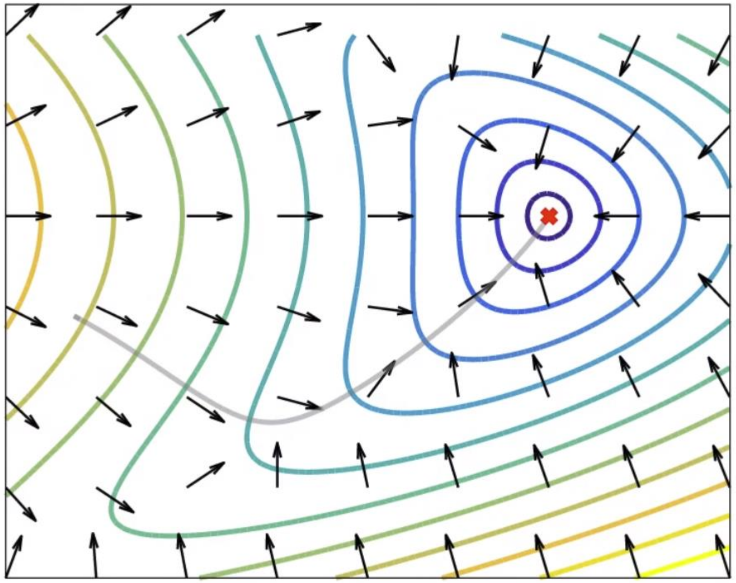
https://www.youtube.com/watch?v=JM7kmWE8Gtc

Today

Optimal Control

Optimal Control: Finding the best control policy for a desired goal

Closed-Loop Solutions

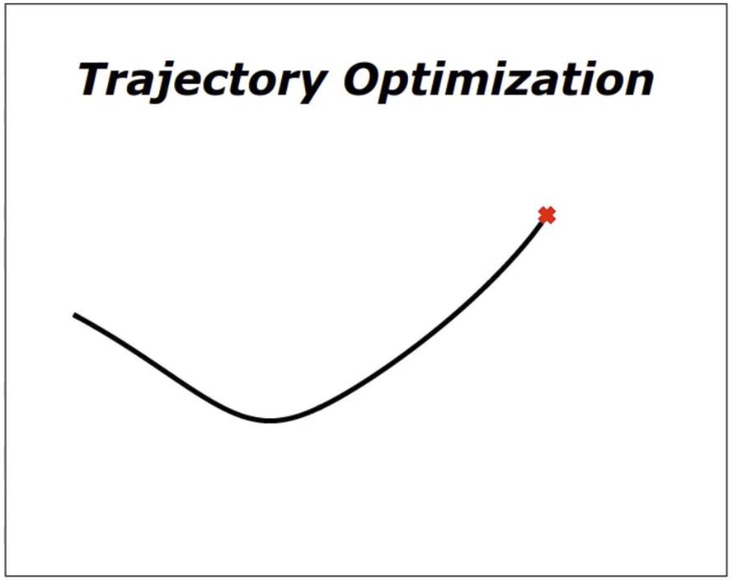


$$u = u(x)$$

"Global Method": Gives action at all states

Very expensive to compute

Open-Loop Solution



$$u = u(t)$$

"Local Method": Gives action at relevant states

Usable in high dimensions

Trajectory Optimization:

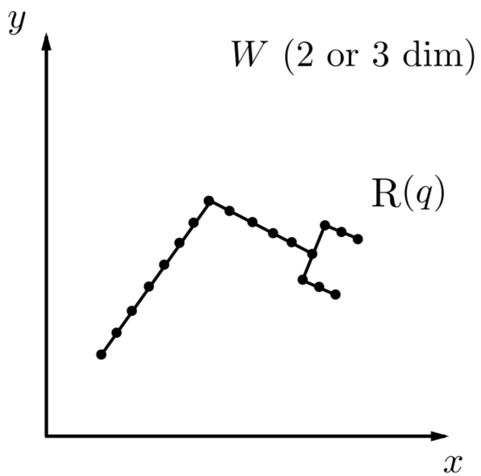
Problem Statement

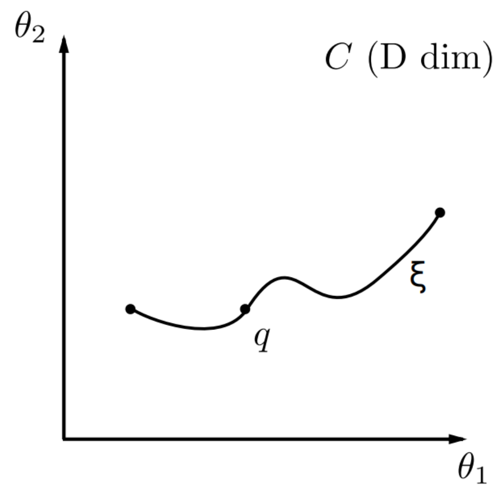
- Trajectory $\xi : t \in [0, T] \rightarrow C$ *Maps time to configurations*
- Objective Functional $U : \Xi \rightarrow \mathbb{R}^{+}$ *Maps trajectories to scalars*
- The objective U encodes traits we want our path to have
  - Path length
  - Efficiency
  - Obstacle avoidance
  - Legibility
  - Uncertainty reduction
  - Human comfort

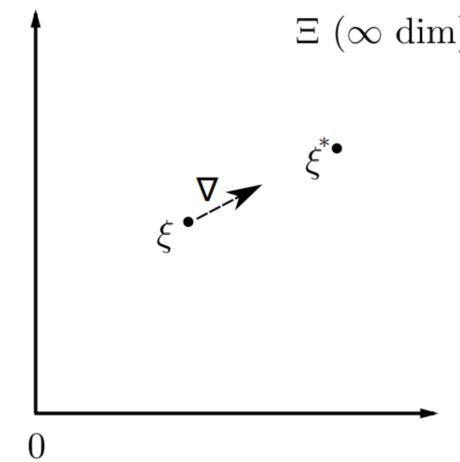
Set of possible trajectories

Goal: Leverage the benefits of randomized sampling with asymptotic optimality

Problem Specification: Spaces







World Space W($\mathbb{R}$^3)

Configuration Space
C($\#DoF$)

Trajectory Space
$\Xi(\infty\ dim)$

Robot pose in World Space (set of points)

Single point in Configuration Space

Trajectory through Configuration Space (set of points)

Single point in Trajectory Space

Problem Specification: Optimization

Trajectory Optimization seeks to find an optimal trajectory ξ^*:

$$\xi^* = argmin_{\{\xi \in \Xi\}} U[\xi]$$

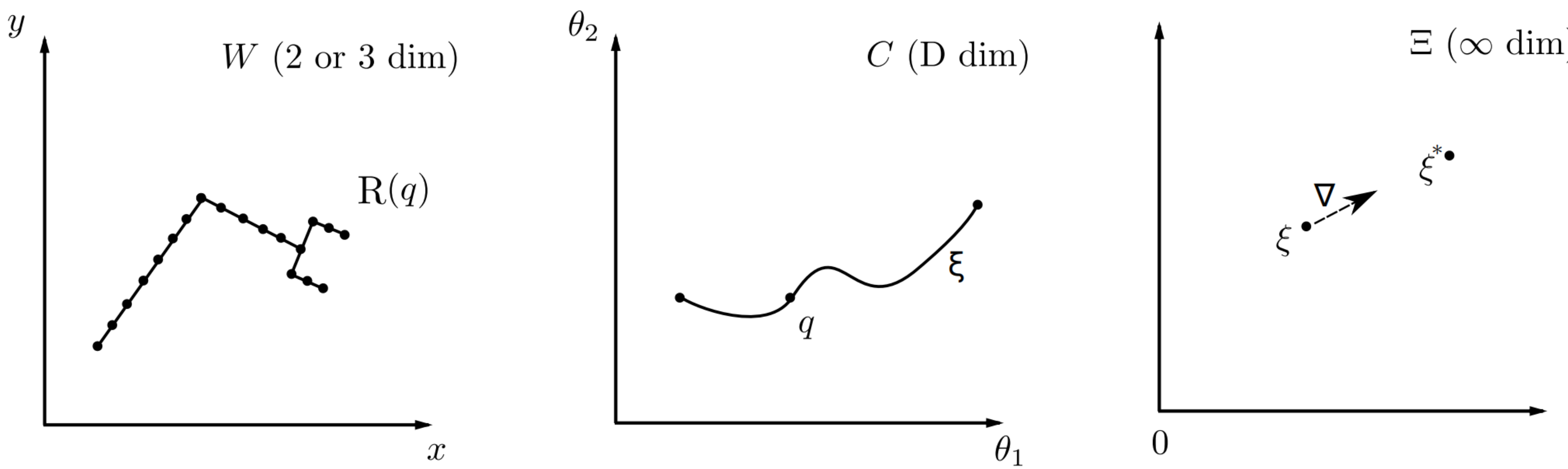
s.t.

$$\xi(0) = q_s$$

$$\xi(T) = q_g$$

…(any other constraints we want)

Problem Specification: Optimization



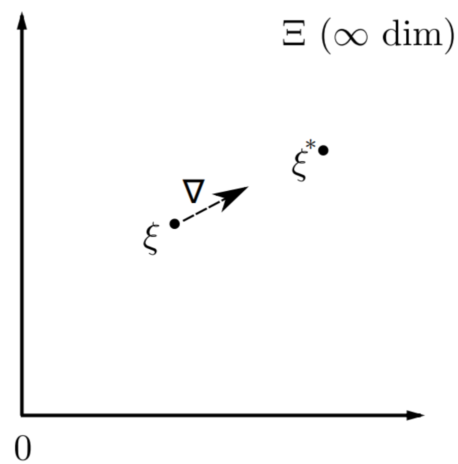
Want to optimize ξ to a global minimum of our objective **U**

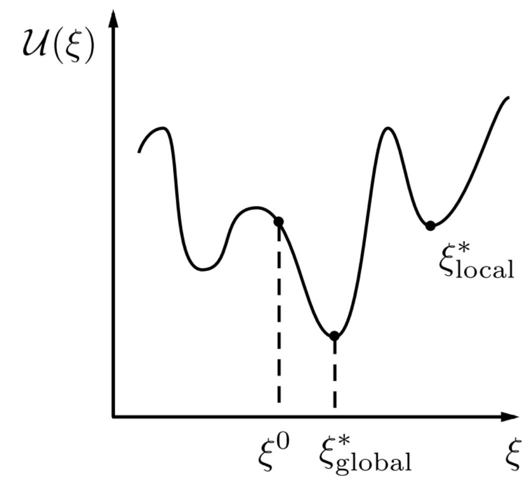
=> Usually too hard!

Instead, optimize ξ to a local minimum of our objective **U**

=> If the solution is bad, resample ξ and try again

Problem Specification: Optimization





Gradient Descent is a popular technique for finding local minima

$$\xi_{t+1} = \xi_t - \frac{1}{\alpha} \nabla_\xi U(\xi_t)$$

Points in the direction of greatest ascent for ξ

Functional Gradient Descent

To compute the derivatives of a functional, some trickery is required:

Ξ is a Hilbert space with Euclidean inner product

$$U[\xi] = \int_0^T F\big(t, \xi(t), \xi'(t)\big) dt$$

$$\nabla_\xi U(t) = \frac{\delta F}{\delta \xi(t)}(t) - \frac{d}{dt}\frac{\delta F}{\delta \xi'(t)}(t)$$

$$\nabla_\xi U(t) \in \Xi$$

Functional Gradient Descent

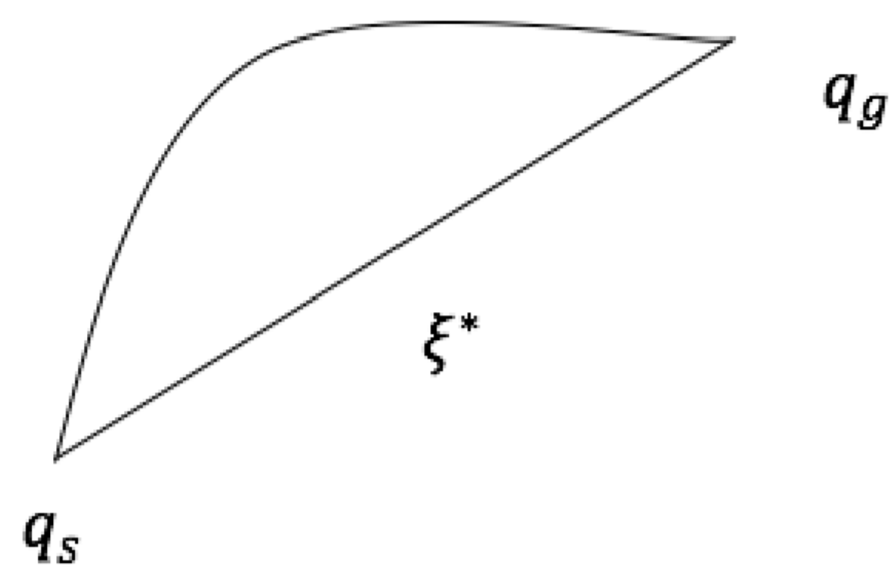
Example: Cost function to minimize magnitude of velocity

Given $U[\xi] = \frac{1}{2}\int_0^T \|\xi'(t)\|^2 dt$

An optimal trajectory will have the following properties:

Shape: Linear.
(Traveling less distance will require less velocity over the time interval.)

Timing: Constant velocity



Making Trajectory Optimization Useful

Need to provide a good choice for $\mathbf{U}[\xi]$.

CHOMP: Covariant Hamiltonian Optimization for Motion Planning

Uses a cost function $\boldsymbol{U[\xi] = U_{smooth}[\xi] + \lambda U_{obs}[\xi]}$

Smoothness cost: $\boldsymbol{U_{smooth}[\xi] = \frac{1}{2}\int_0^T ||\xi'(t)||^2\, dt}$

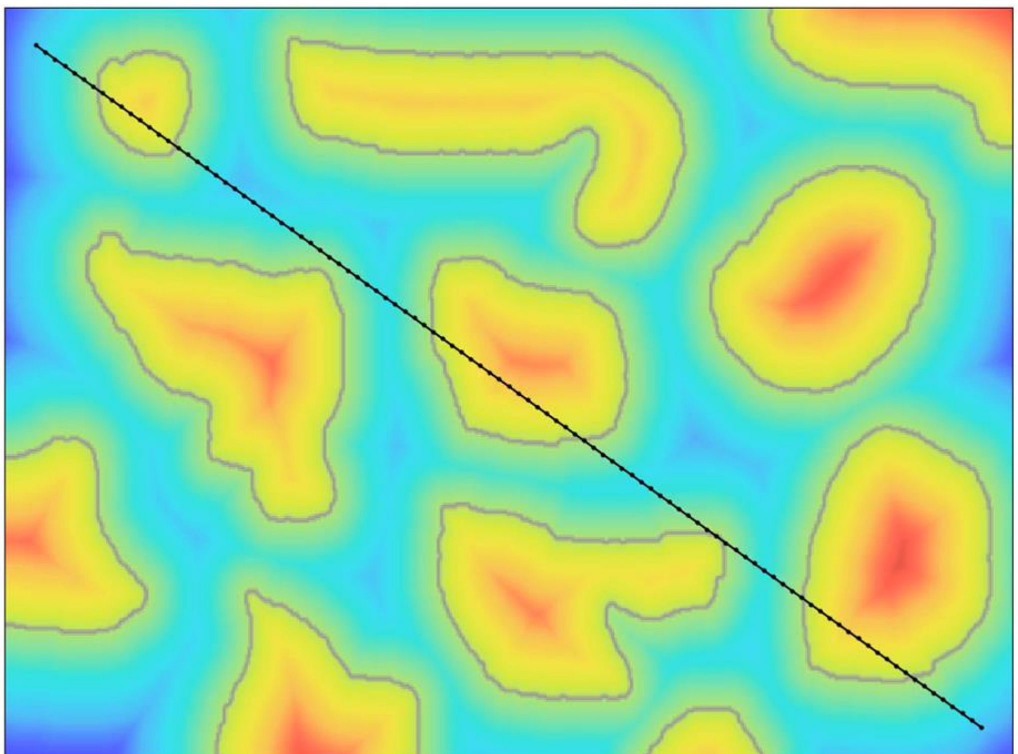
Obstacle cost: $\boldsymbol{U_{obs}[\xi] = \int_t \int_u c\left(\phi_u(\xi(t))\right) * \left\| \frac{d}{dt}\phi_u(\xi(t)) \right\| du dt}$

Cost function that computes distance to closest obstacle	Forward Kinematics function that computes location of robot body point u at time t in ξ	Norm of the velocity for body point u at time t in ξ

Making Trajectory Optimization Useful

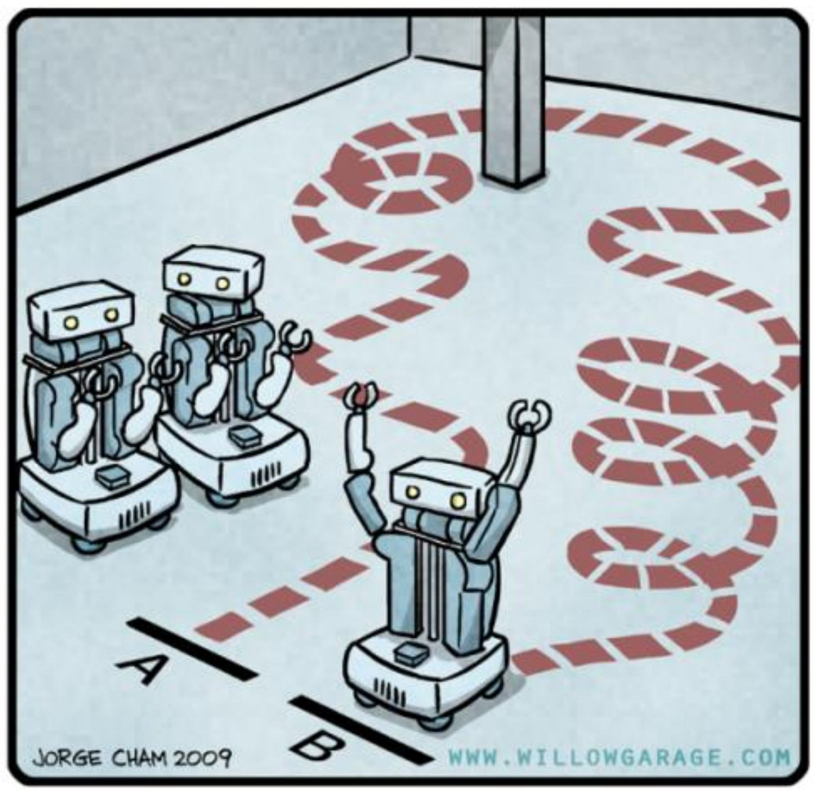
CHOMP: Covariant Hamiltonian Optimization for Motion Planning

Uses a cost function $\boldsymbol{U}[\xi] = \boldsymbol{U_{smooth}}[\xi] + \boldsymbol{\lambda U_{obs}}[\xi]$



A-HRI: More Open Problems

R.O.B.O.T. Comics



"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

Robots Asking for Help

Asking for Help Using Inverse Semantics

Stefanie Tellex, Ross A. Knepper

Adrian Li, Thomas M. Howard, Daniela Rus, Nicholas Roy

Supported in part by the Boeing Company, and in part by the U.S Army Research Laboratory under the Robotics Collaborative Technology Alliance.

Massachusetts Institute of Technology

UNIVERSITY OF CAMBRIDGE

CSAIL

Learning by Demonstration

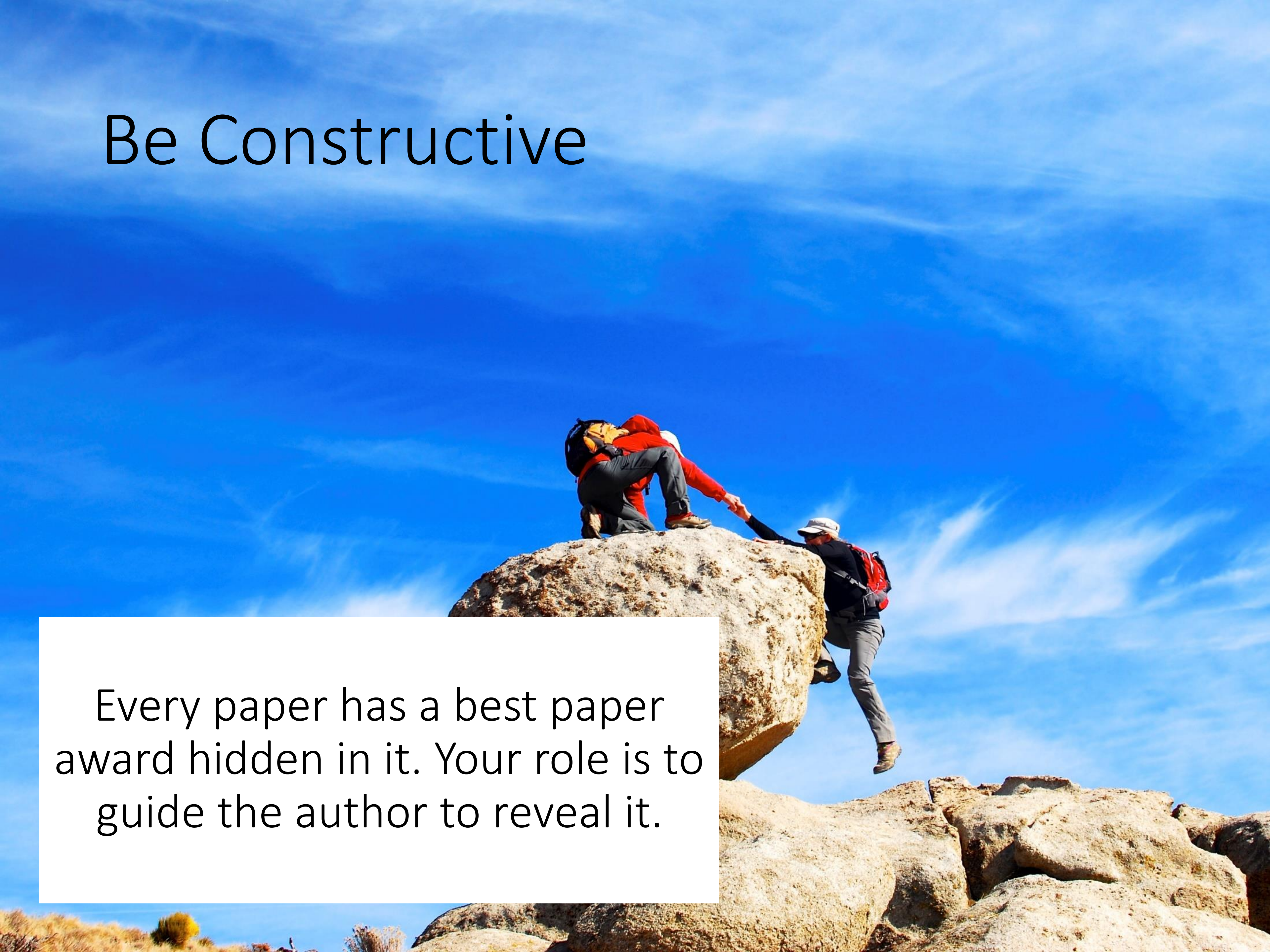
Parsing Natural Language Instruction

Reviewing Papers

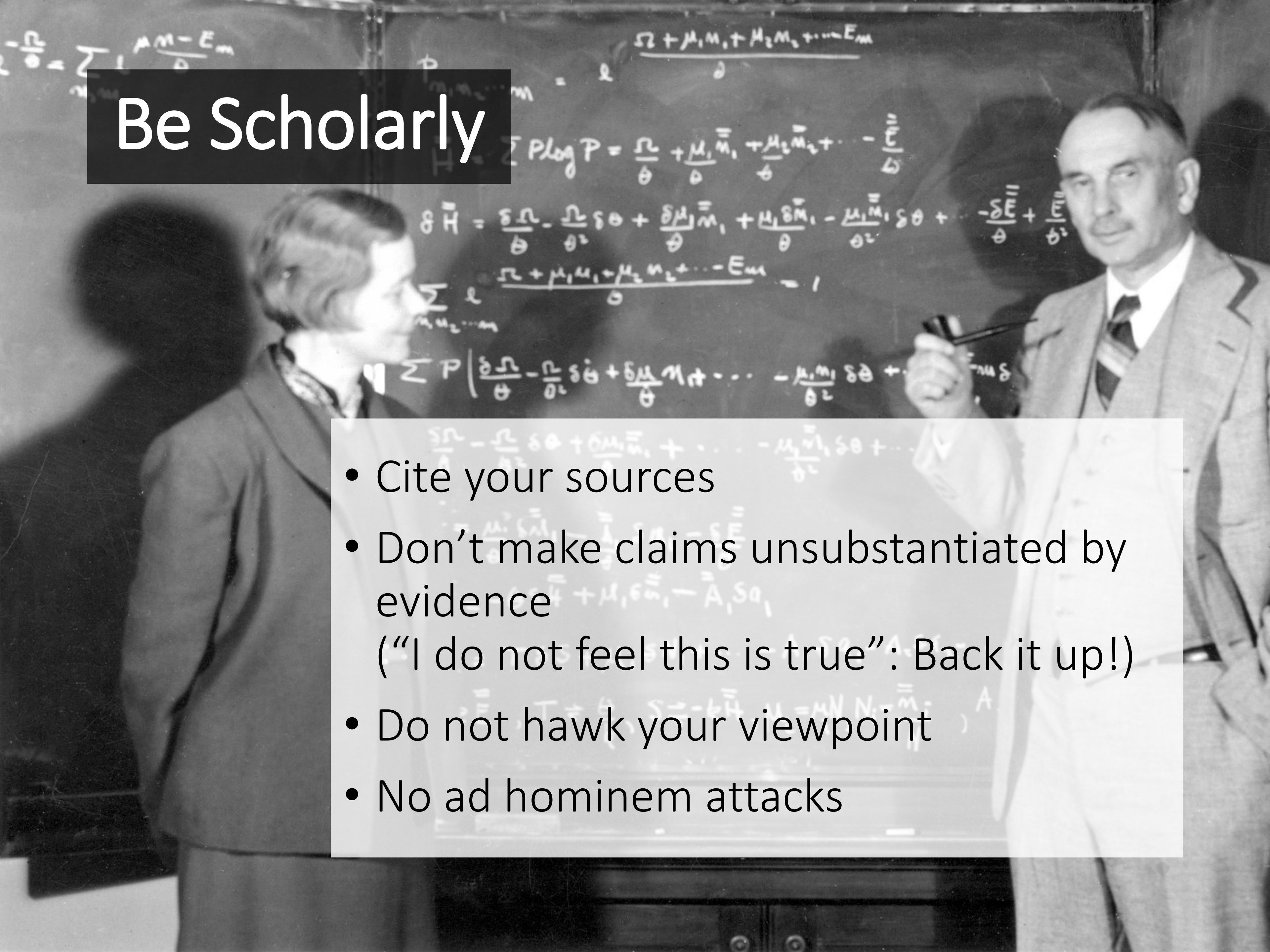
Be Compassionate

Invert your position and ask yourself how you'd feel if you received your review. If it makes your blood boil, take a break, and revise your review.

Be Constructive

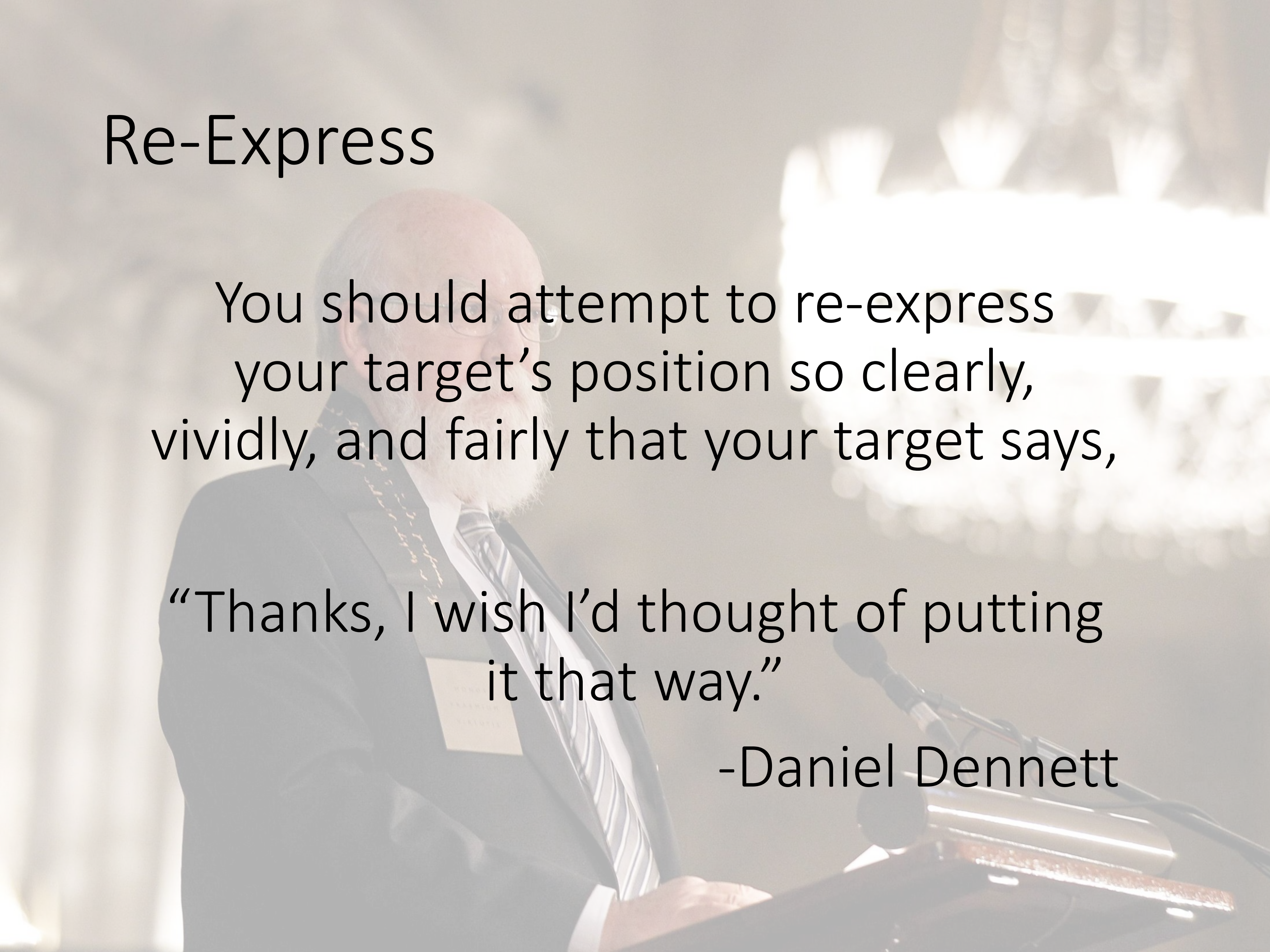


Every paper has a best paper award hidden in it. Your role is to guide the author to reveal it.



Be Scholarly

- Cite your sources
- Don’t make claims unsubstantiated by evidence
  (“I do not feel this is true”: Back it up!)
- Do not hawk your viewpoint
- No ad hominem attacks



Re-Express

You should attempt to re-express your target's position so clearly, vividly, and fairly that your target says,

"Thanks, I wish I'd thought of putting it that way."

-Daniel Dennett

Summary

- Re-express target's position clearly, vividly, and fairly
- List any points of agreements
- Mention anything you have learned from the target
- Professionally introduce rebuttal or critique

Presenting Papers

- Identify key insights
- Identify corner cases
- Identify scaling issues
- Identify unrealistic application constraints
- Avoid being procedural

Presenting a Paper

1. Bigger picture
2. Motivation
3. Problem Statement
4. Why is this hard
5. Key insight
6. Details
7. Results
8. Broader picture
9. Restate the key insight that addressed the problem

Papers for Thursday 1/31:

Need 1 PRO (10m) and 1 CON (5m) speaker each

E-mail Bradley.Hayes@Colorado.edu to sign up

Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective
Baris Akgun et al.

Planning human-aware motions using a sampling-based costmap planner
Jim Mainprice et al.

Project Brainstorming!

Platforms


Parrot


KINOVA
MOVO BETA
A MOBILE
MANIPULATOR
PLATFORM
kinovaMOVO.com


sawyer

Project Brainstorming

Form small groups

A few minutes per topic:

Discuss a capability or system and application scenario that would address the topic

Write down your ideas (brief summaries please)

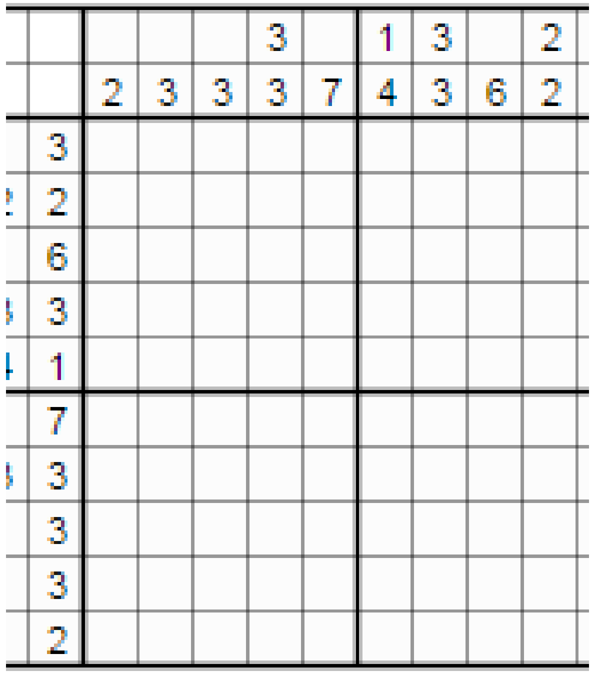
Project Brainstorming

Projects can involve either

building an autonomous system

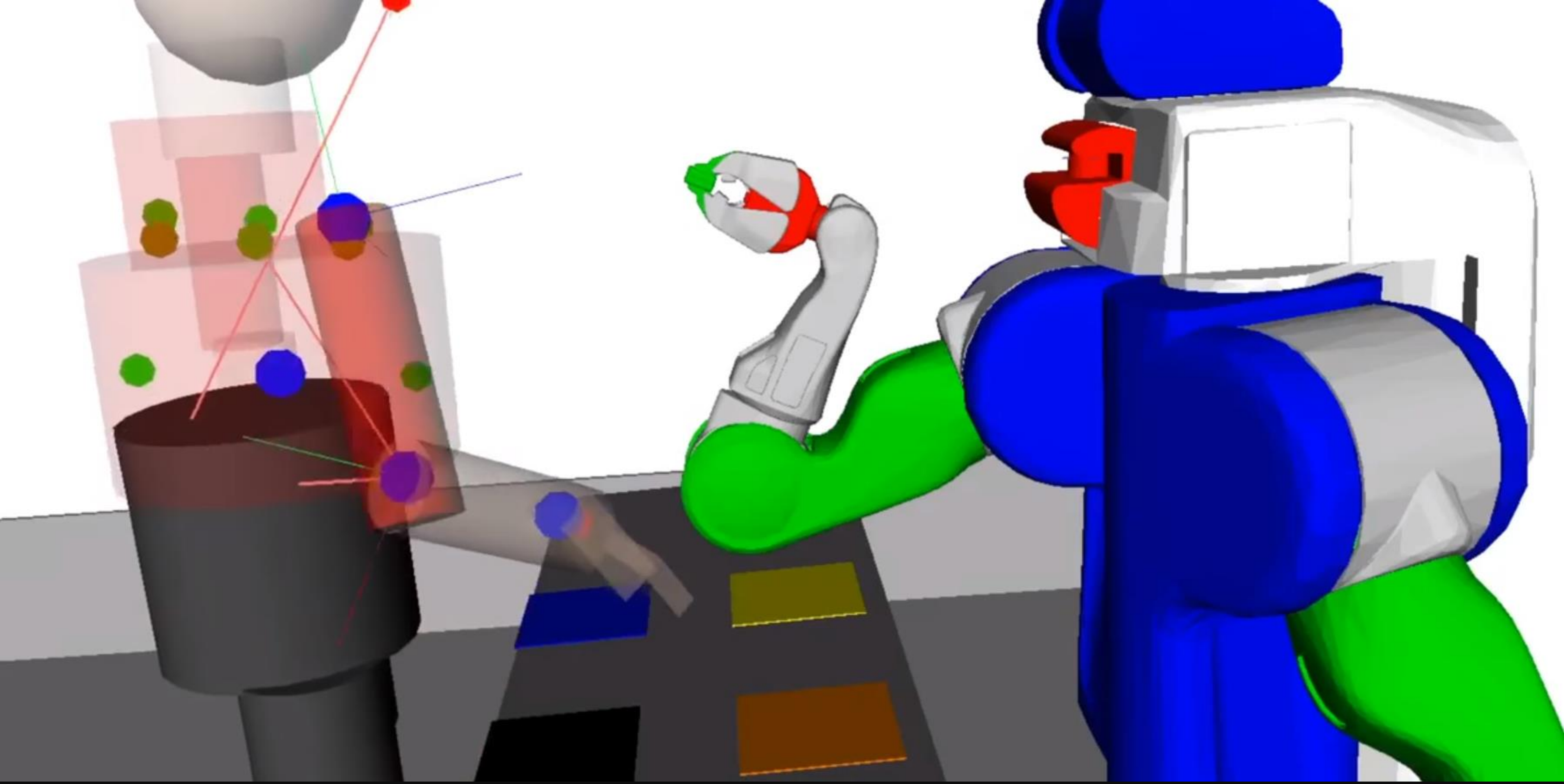
or

testing some aspect of a hypothetical autonomous system



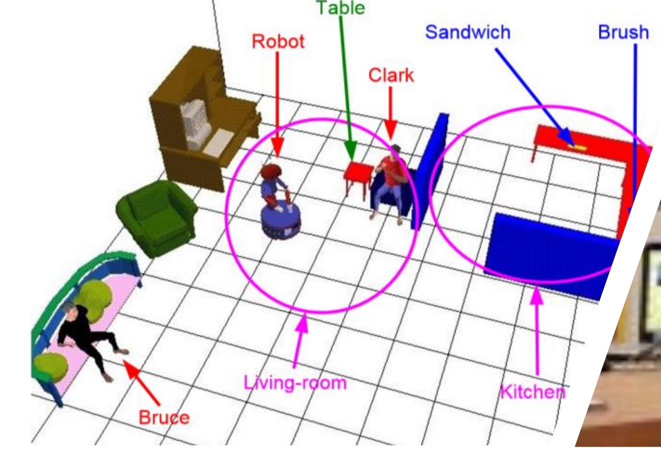
Topic: Personalized Robot Tutoring

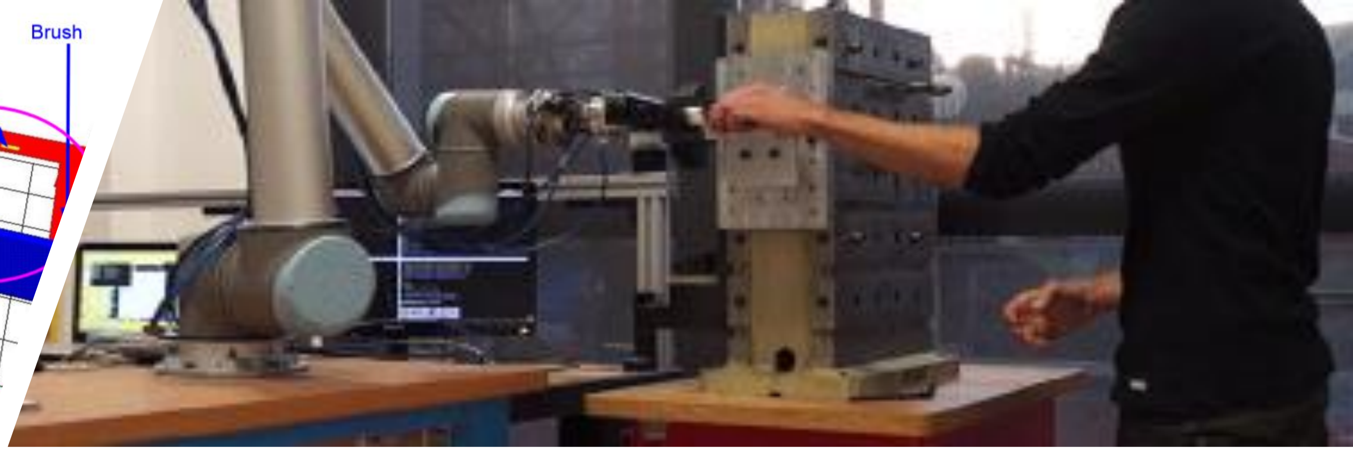
- Language Learning
- Training for Manufacturing (IKEA)
- Nonograms / Games
- Etc.



Topic:
Safe Human-Aware Motion Planning

- Safety systems for mixed human-robot environments
- Visualizations for communicating intent
- Motion planning cost functions incorporating human activity





Topic:
Reducing Human's Uncertainty During Collaboration

Modeling which objects in scene the human is likely to (not) know the locations of

Communicating the robot's plan through pantomime or speech