L1: Introduction

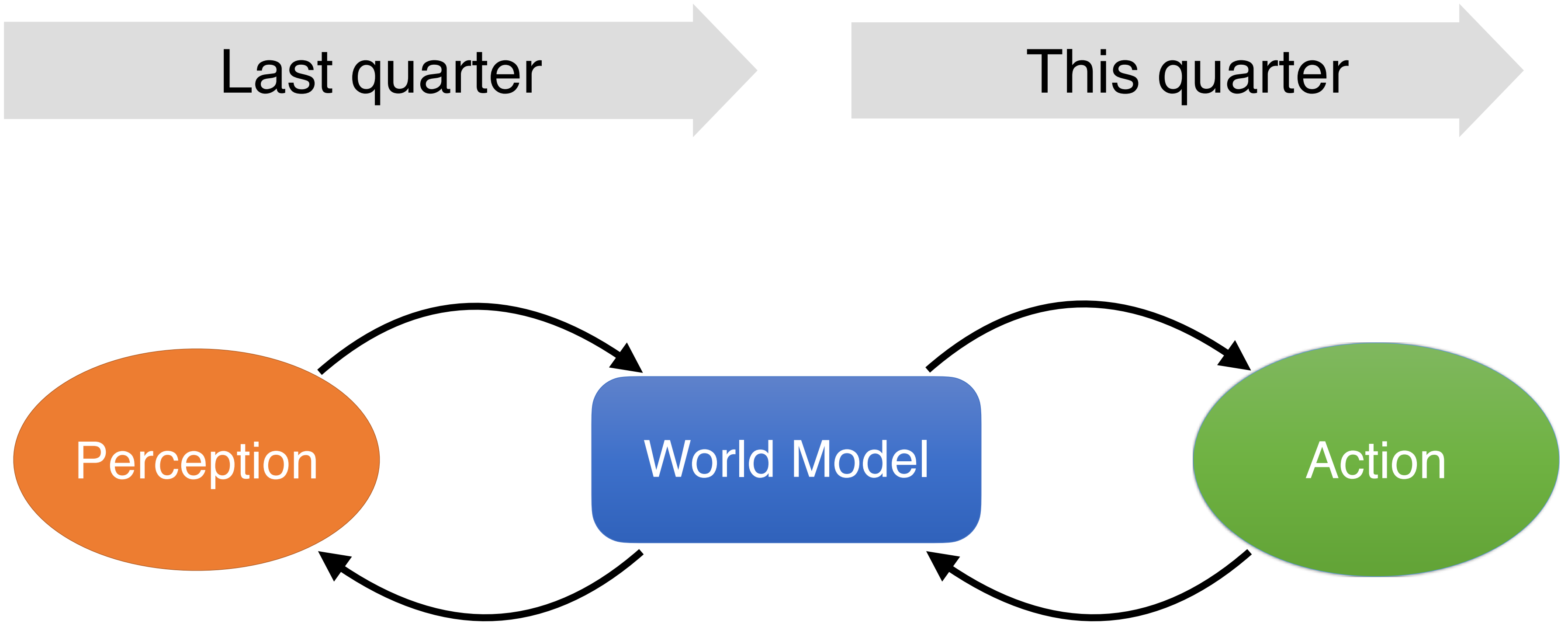
Hao Su

- https://www.youtube.com/watch?v=fn3KWM1kuAw

Agenda

- Syllabus
- Logistics
- $\mathbb{SO}(3)$

Syllabus


Last quarter
This quarter
Perception
World Model
Action

Vision → Robotics

Passive AI

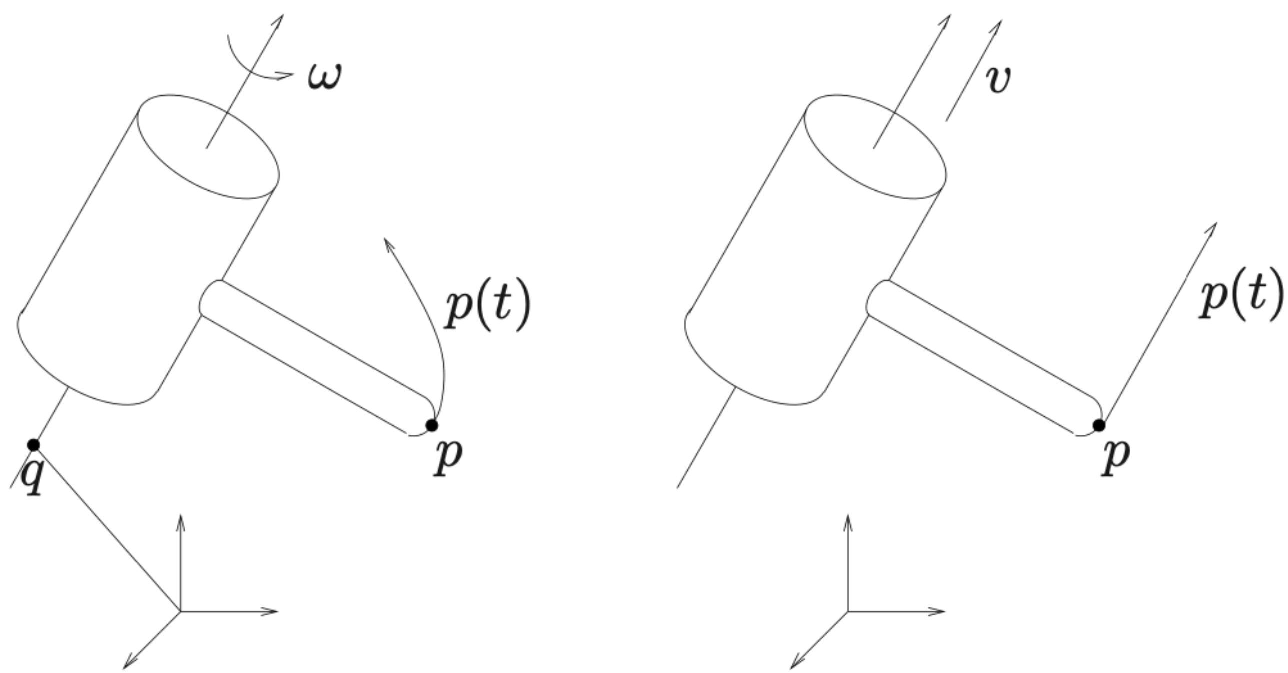
- We know how to fit data well (by “deep learning”)
  - e.g., computer vision, natural language processing

Vision → Robotics

- We aspire that autonomous agents can perform tasks and “grow” through interaction experiences
  - Need the ability to **interact**

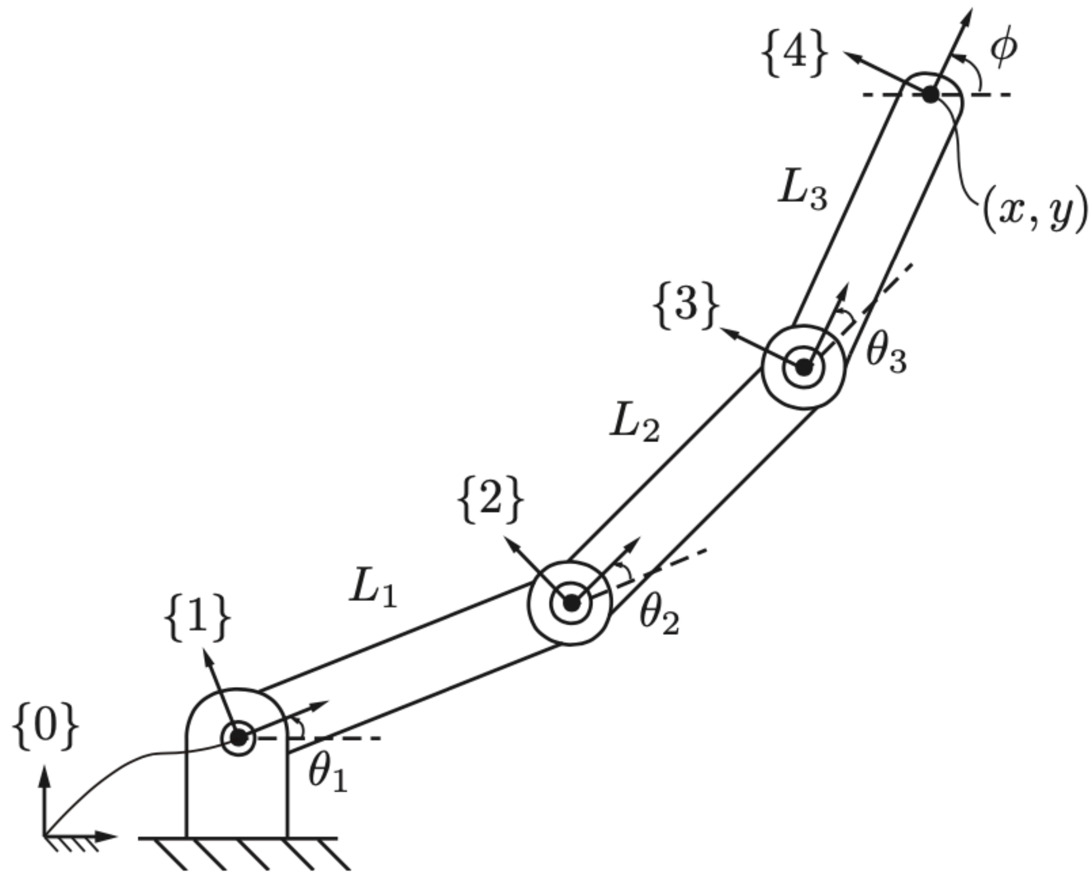
Topics Covered in This Course

- Modeling Robots by Rigid-Body Geometry



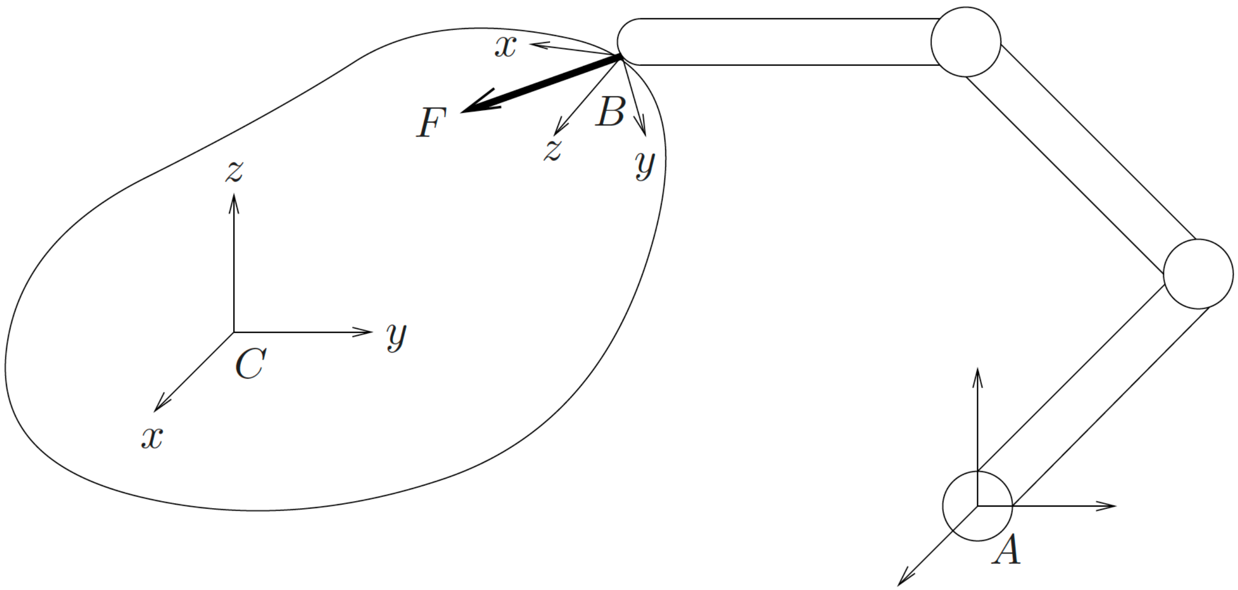
Topics Covered in This Course

- Forward and Inverse Kinematics of Robots



Topics Covered in This Course

- Generalized Force and Inertia



Topics Covered in This Course

- Friction, Contact Model, and Grasp

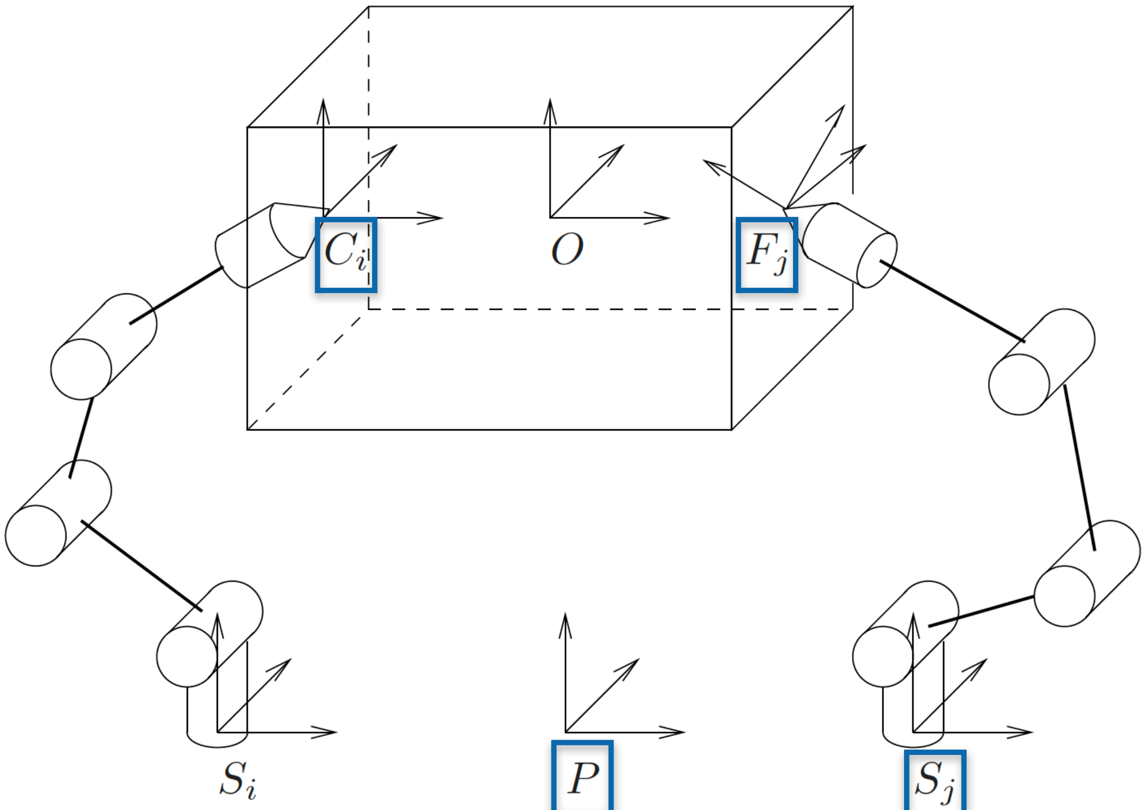
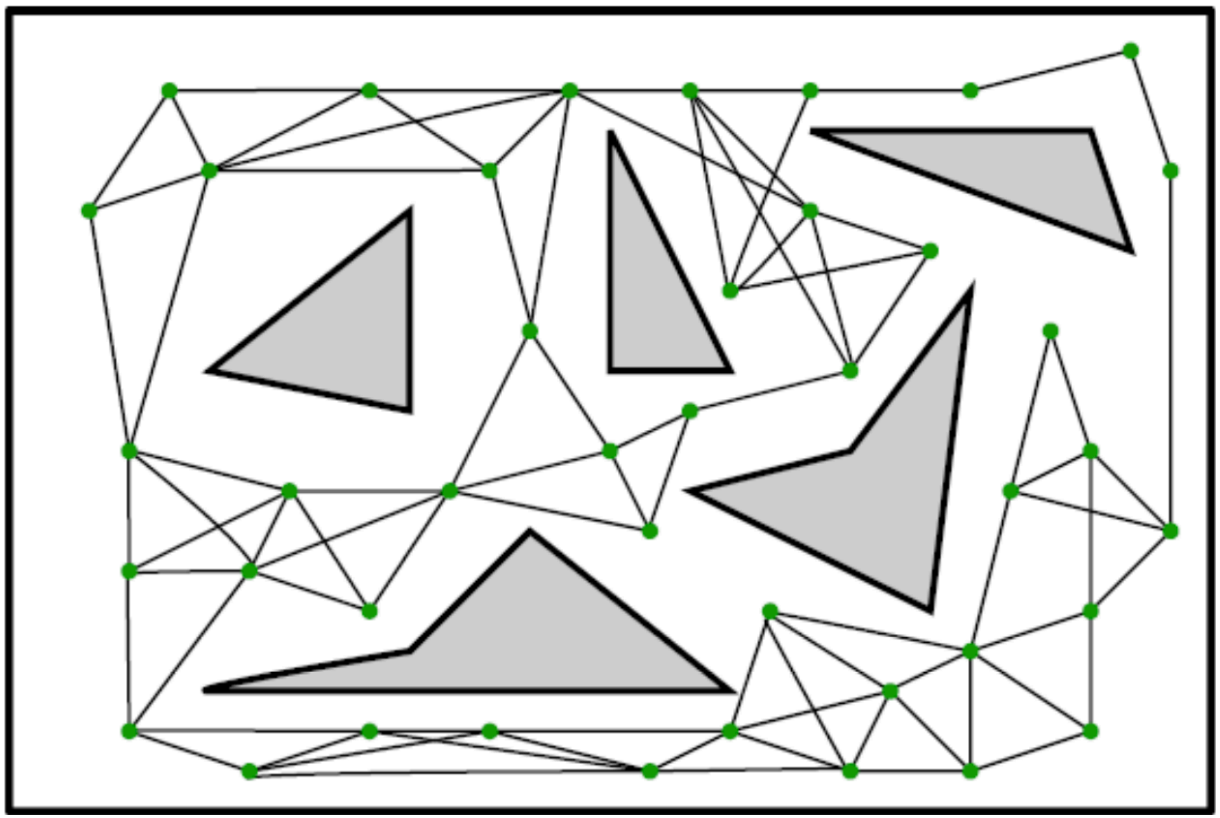


Figure 5.14: Grasp coordinate frames.

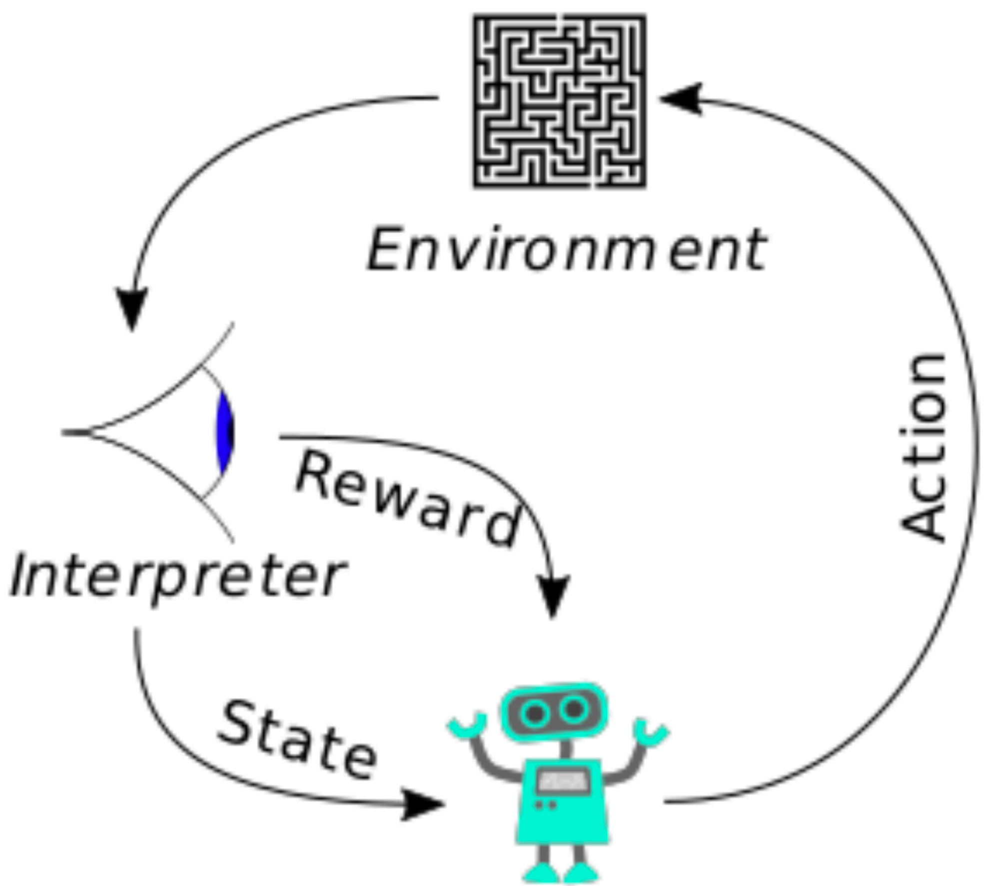
Topics Covered in This Course

- Classical Planning and Control



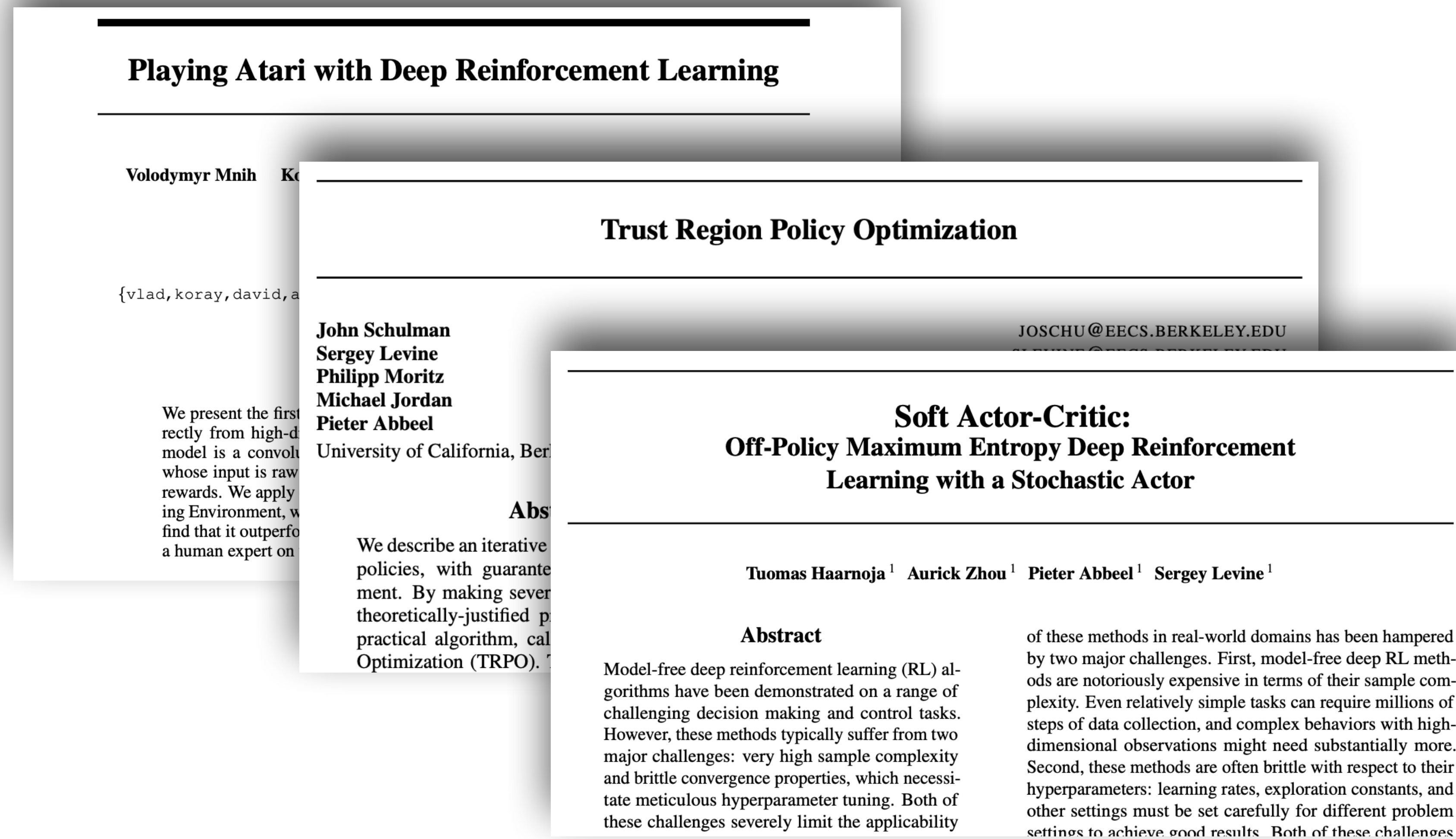
Topics Covered in This Course

- Concepts of Reinforcement Learning



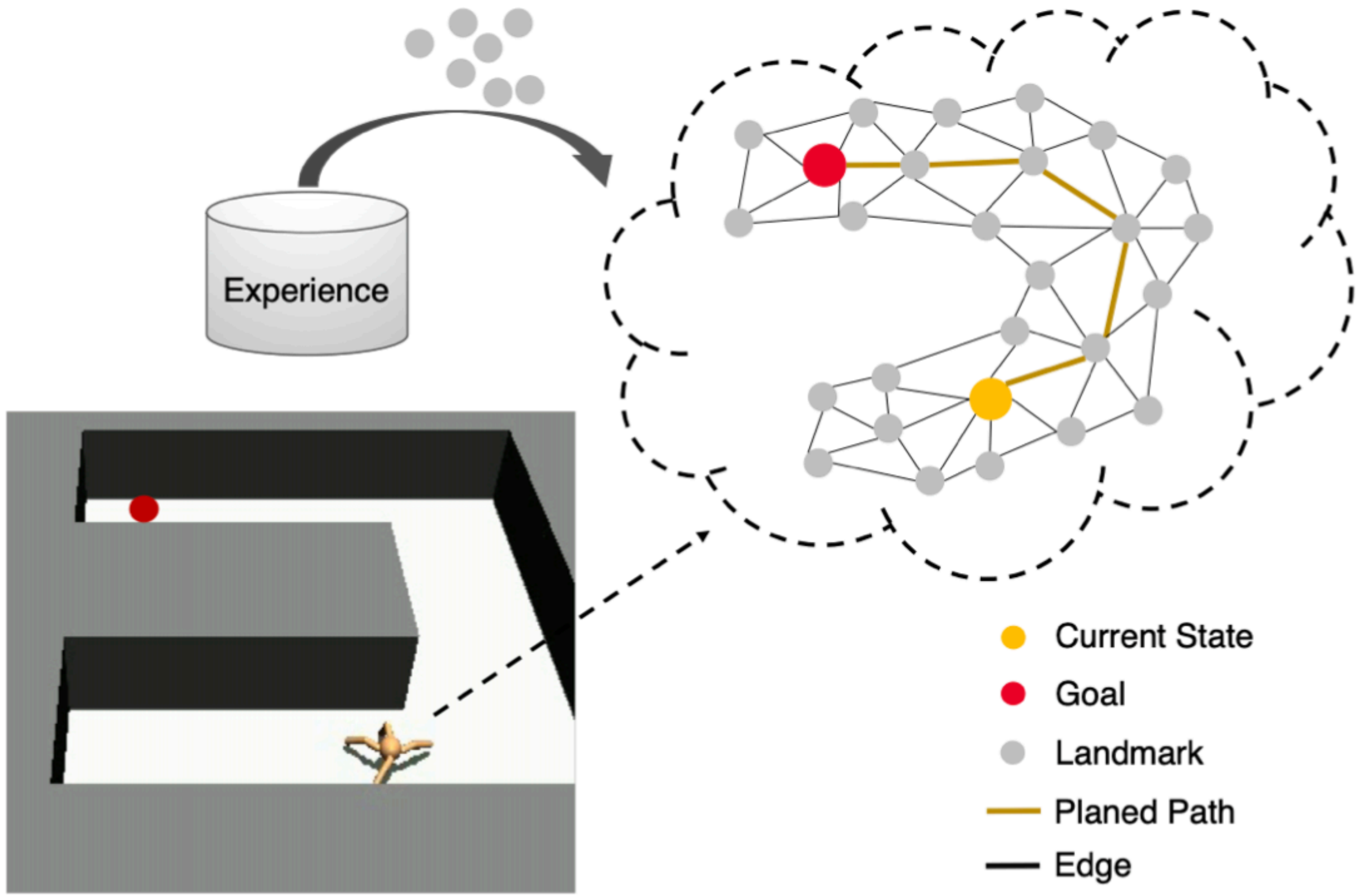
Topics Covered in This Course

- Deep RL Frameworks



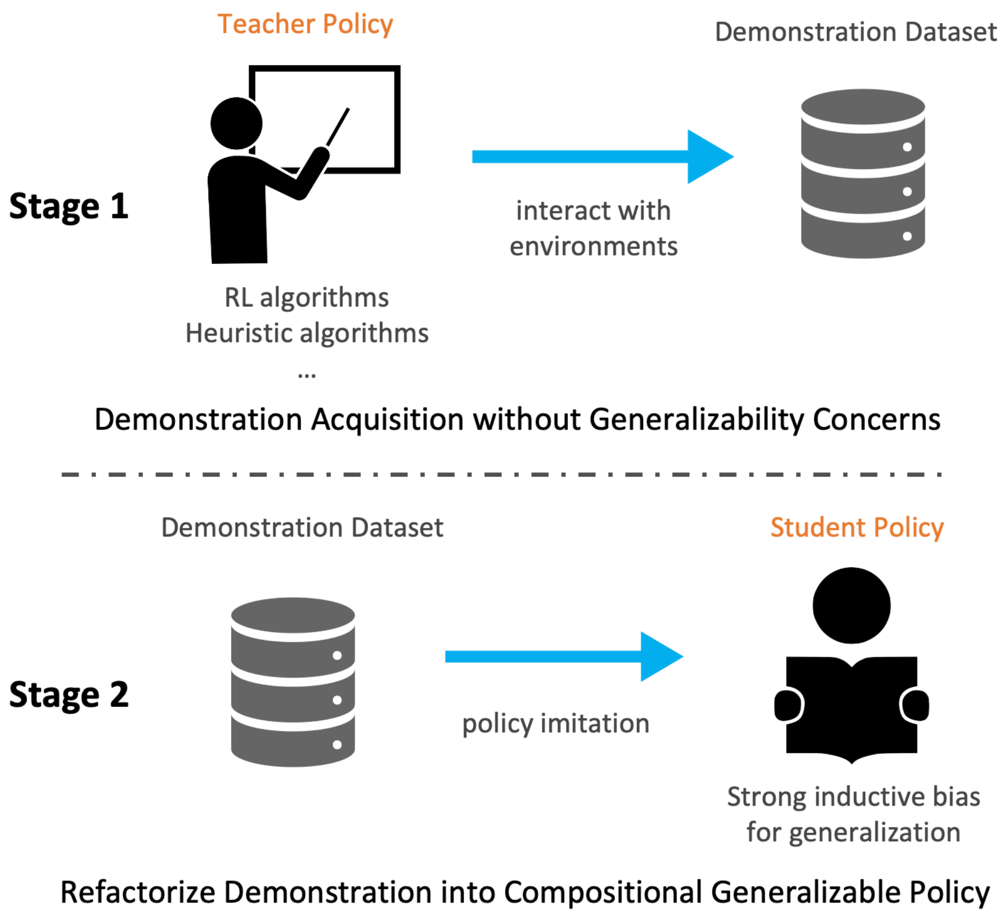
Topics Covered in This Course

- Hierarchical RL



Topics Covered in This Course

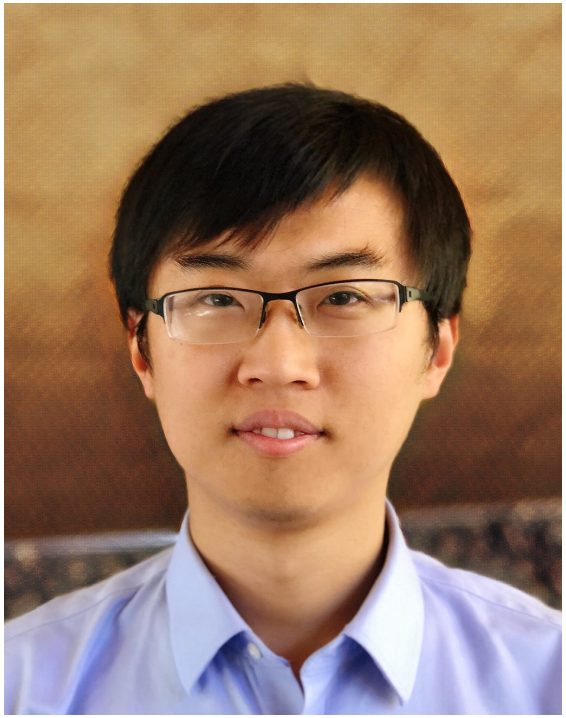
- Generalizability of RL



Course Logistic

Instructors

Instructor: Hao Su



TA: Zhiao Huang

Teaching Goal

- Foundational
  - Programming problems ask you to **implement low-level modules from scratch**

- Hands-on
  - **Heavy** programming assignments to exercise what are taught in class

Pre-requisite: Technique

- **Skilled** in Linear Algebra, Multi-variable Calculus, and Deep Learning
- **Familiar** with Probability and Numerical Methods
- **Strong** programming skills
  - Familiar with Linux Toolchain
  - Familiar with python, numpy, and pytorch
- Course/project experiences in deep learning

Background Check

- On Piazza now (HW0)
  - Visible to enrolled and waitlist students
- 5 points in your final grade
- **Mandatory!** We will not grade your subsequent homeworks without seeing your HW0.
- If you are in the waitlist and intend to enroll, you need to submit HW0 by this deadline
- Due: 04/03/2022

Pre-requisite: Resources

- This course requires deep learning resources (to run reinforcement learning challenges)

- Unfortunately, we do not have computational resources to support ~50 students

- Please find the server with the following configuration:
  >= 50G disk space
  >= 1 GPU for deep learning

Assignments

- Weekly homeworks
  - Can drop one homework without penalty

- Final project: final week

- No mid-term/final exams

- Extra credit for participation 5% (ask/answer questions in class, attend office hours)

Assignments

- Practice basic concepts and algorithms; build individual modules

- Final project: integrate modules and test new ideas. Score by performance ranking. Online evaluation system will be set up.

- We estimate **>=15 hrs per week** (out of class) solid time commitment

- We allow you to see homework (through Piazza) and attend the competition *even if you **audit** the course*

Course Resources

- Course website: https://haosulab.github.io/ml-for-robotics/SP22/index.html (Google "Hao Su" → Prof. Homepage → Teaching → this link)
  - Collaboration policy
  - Lecture slides
  - Office hour and location
- Piazza
  - Homework/Solution release
  - Discussions

Office Hour

- Check course website

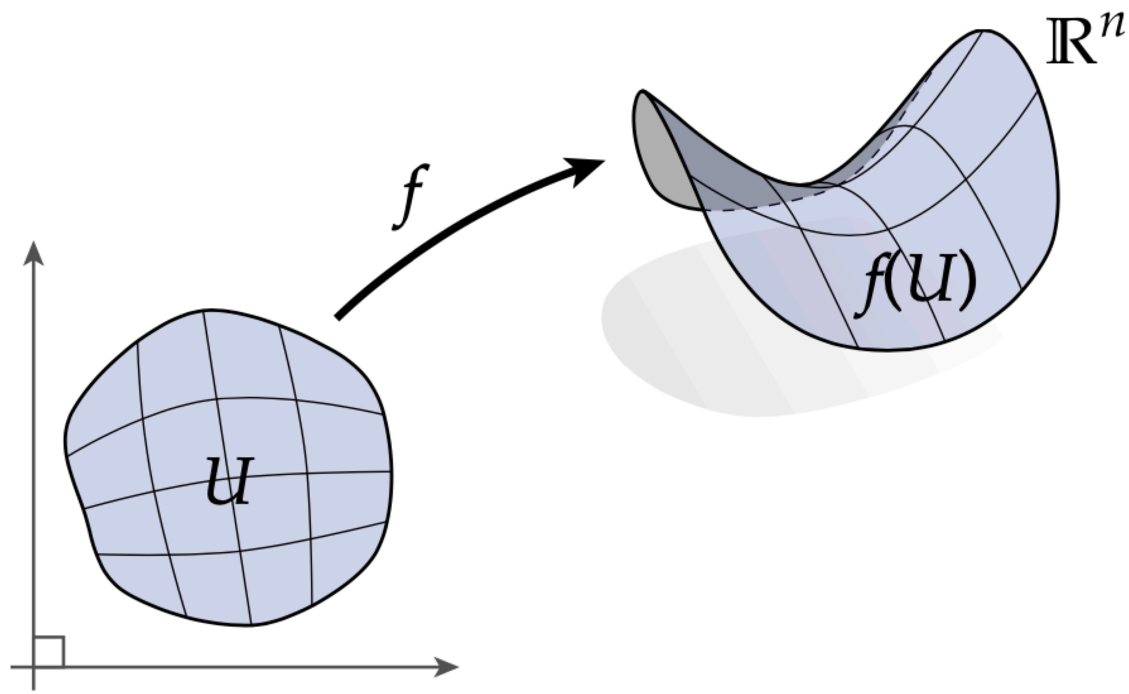
Questions?

Concepts of Differential Geometry

Parameterized Surfaces

A **parameterization** is a map from the domain U into $\mathbb{R}^n$

$$f : U \to \mathbb{R}^n$$



The set of points $f(U)$ is called the **image** of the parameterization.

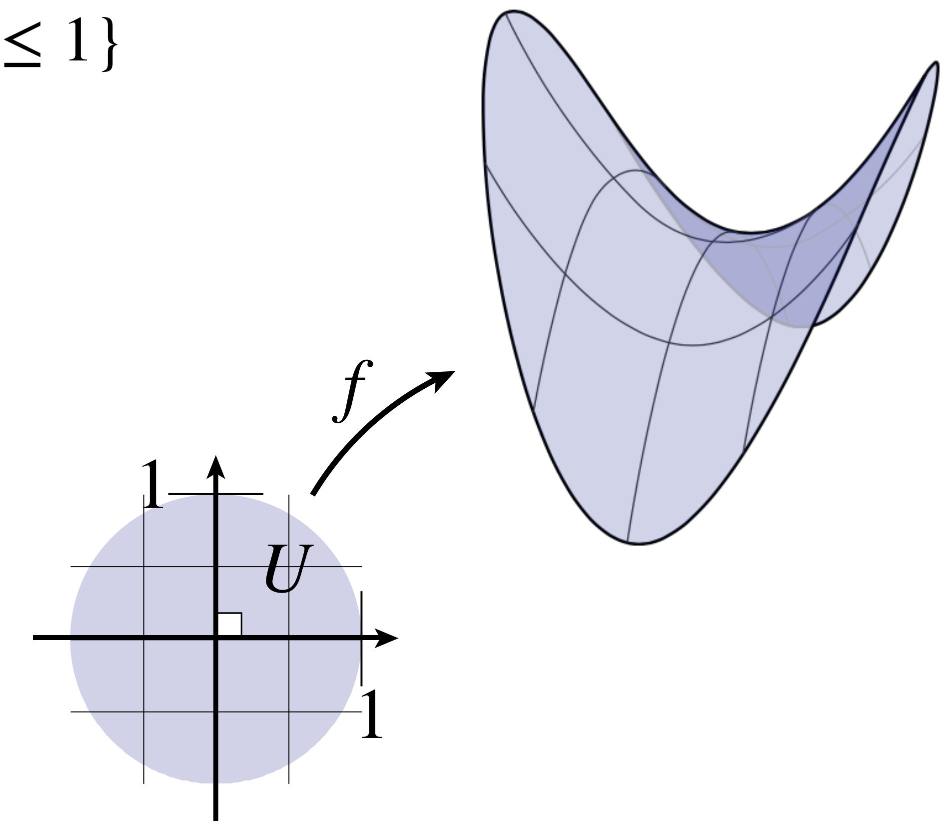
Image from Wikipedia

Example

- Example: We can express a *saddle as a parameterized* surface*:*

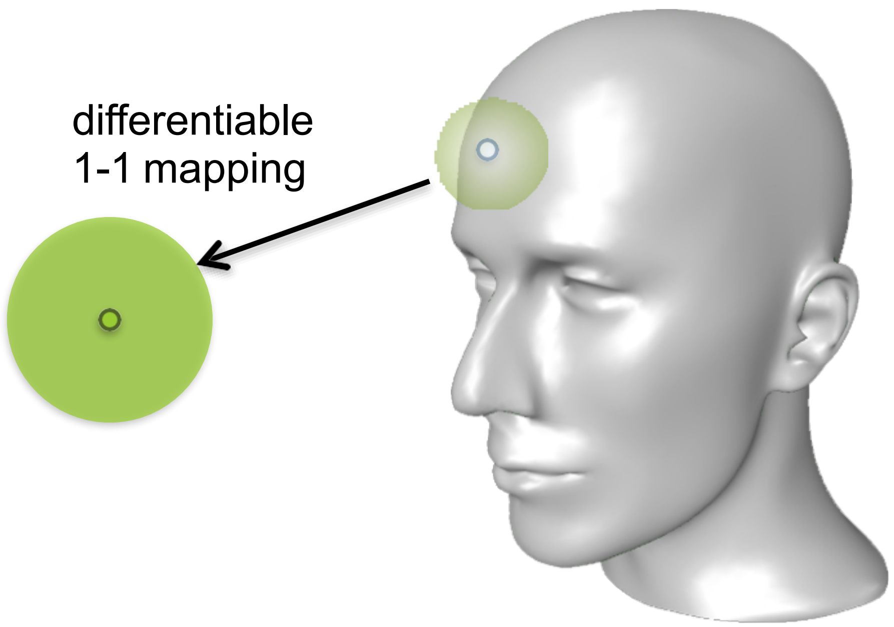
$$U := \{(u, v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\}$$

$$f(u, v) = [u, v, u^2 - v^2]^T$$



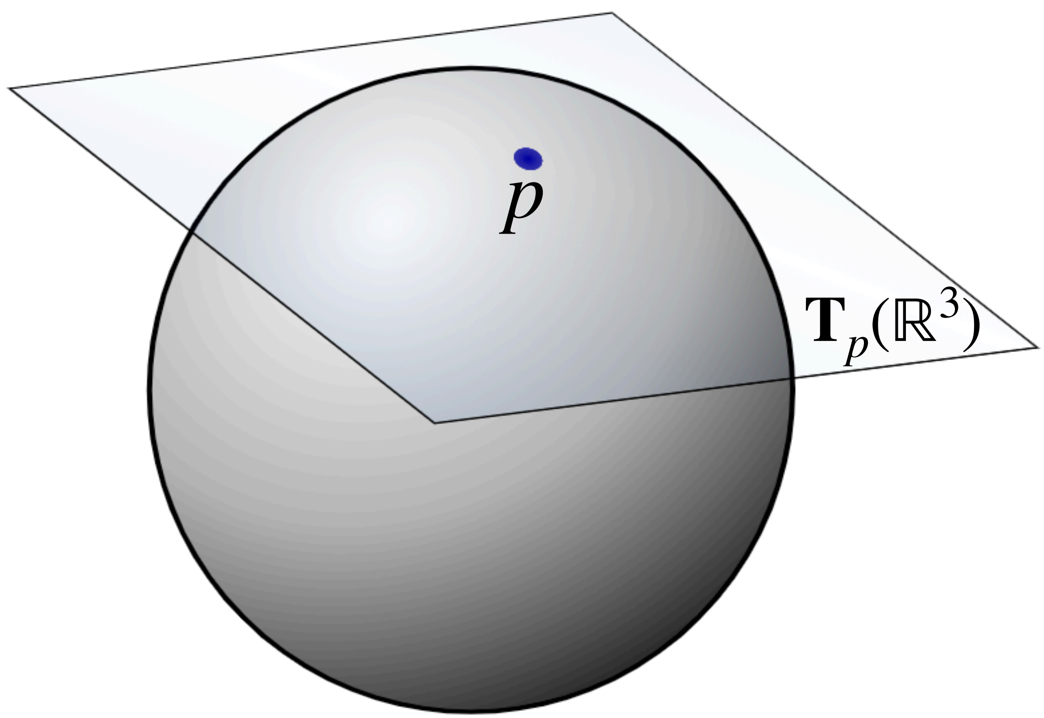
Manifold

- Things that can be discovered by local observation: point + neighborhood



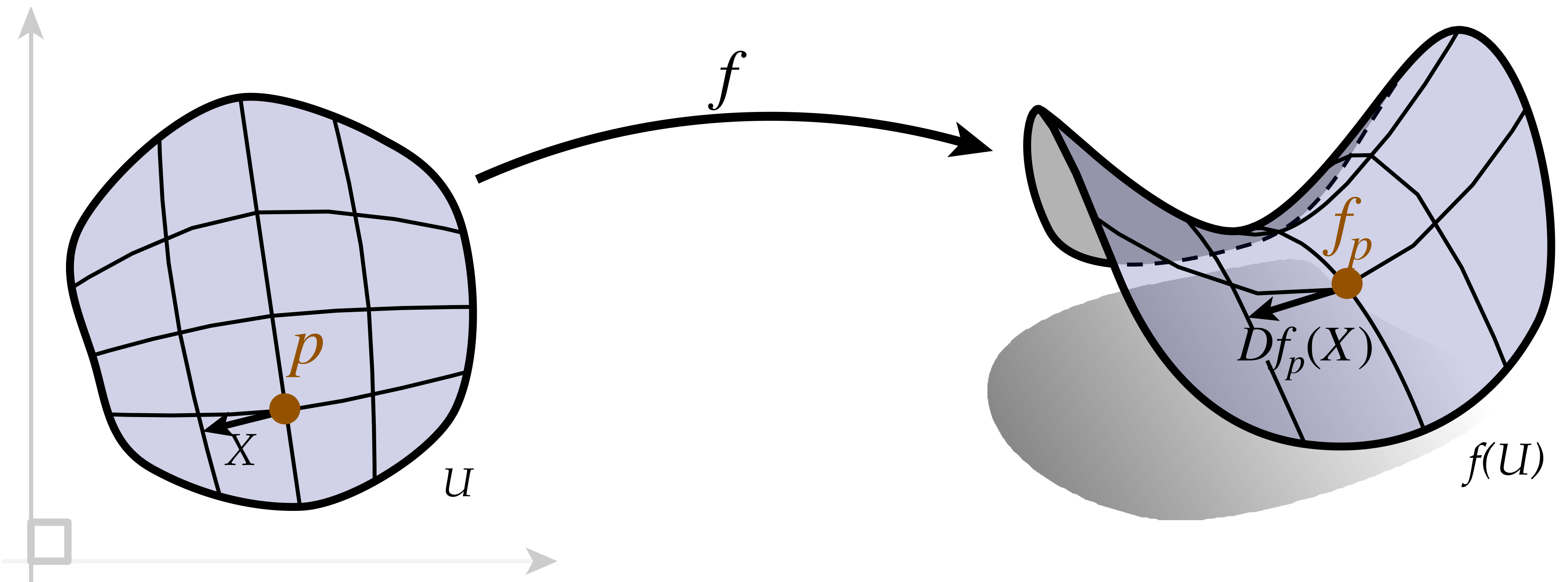
Tangent Plane

- One can attach to every point p a tangent plane $\mathbf{T}_p$
- Intuitively, it contains the possible directions in which one can tangentially pass through p.

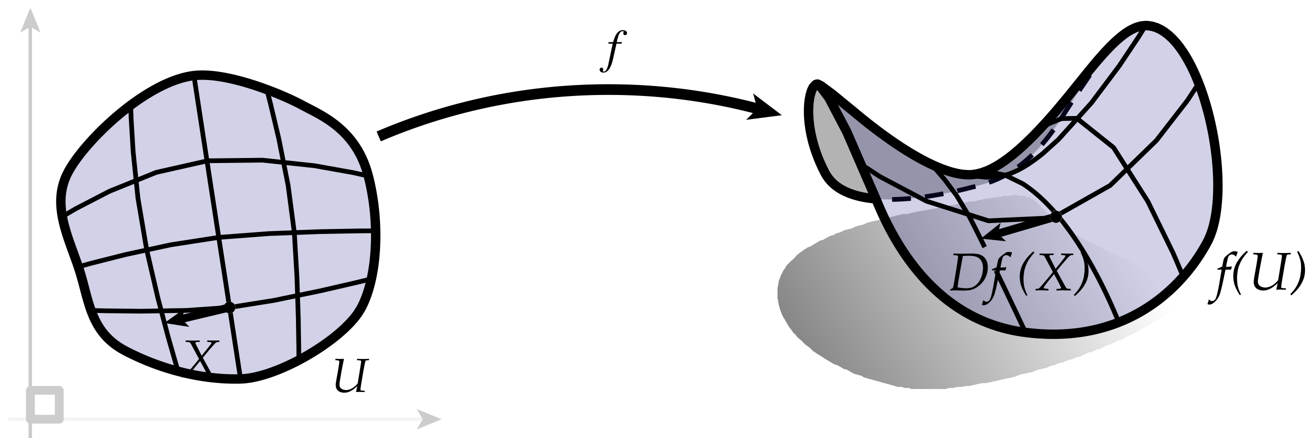


Differential of a Surface

- Relate the movement of point in the domain and on the image



Differential of a Surface

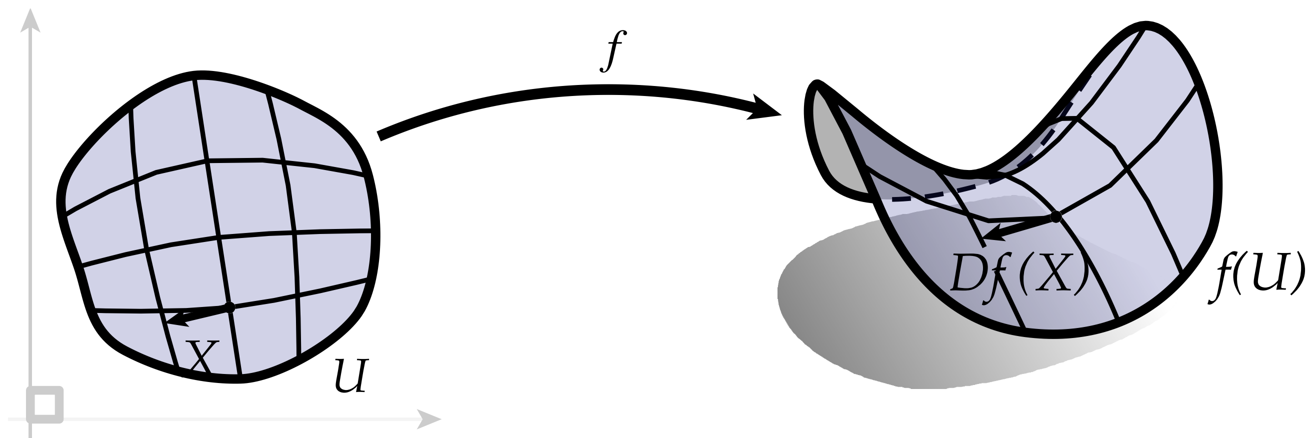


Total differential: $df = \frac{\partial f}{\partial u}du + \frac{\partial f}{\partial v}dv \Longrightarrow \Delta f \approx \frac{\partial f}{\partial u}\Delta u + \frac{\partial f}{\partial v}\Delta v$

If point $p \in \mathbb{R}^2$ moves along vector $X = [u, v]^T$ by ϵ, the movement of f_p is:

$$\Delta f_p \approx \frac{\partial f}{\partial u}(\epsilon u) + \frac{\partial f}{\partial v}(\epsilon v) = \epsilon \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right] \begin{bmatrix} u \\ v \end{bmatrix}$$

Differential of a Surface



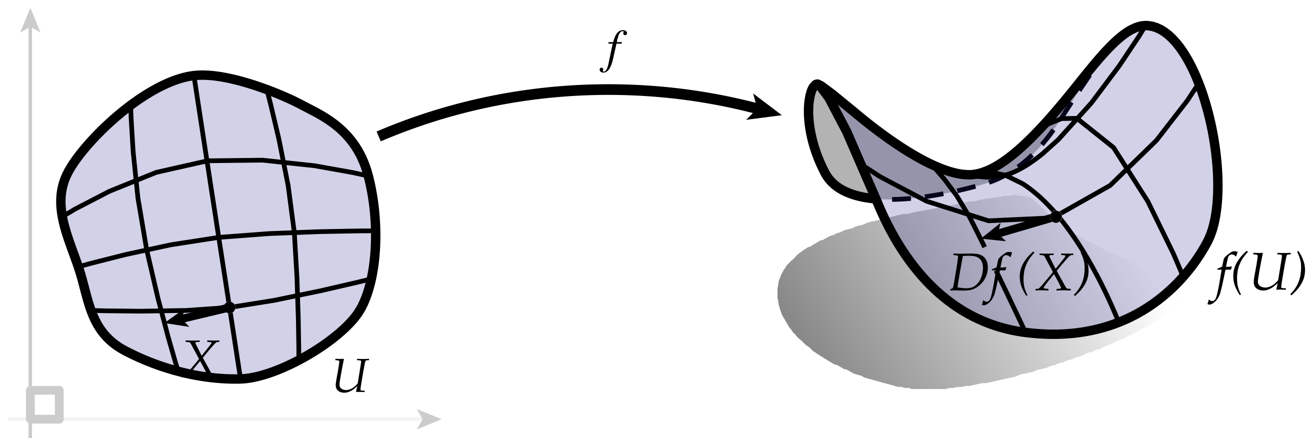
Total differential: $df = \frac{\partial f}{\partial u} du + \frac{\partial f}{\partial v} dv$

If point $p \in \mathbb{R}^2$ moves with velocity $X = [u, v]^T$ by ϵ, the movement of f_p is:

$$\Delta f_p \approx \frac{\partial f}{\partial u}(\epsilon u) + \frac{\partial f}{\partial v}(\epsilon v) = \epsilon \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right] \begin{bmatrix} u \\ v \end{bmatrix}$$

$$Df_p := \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right] \in \mathbb{R}^{3 \times 2}$$

Differential of a Surface



Total differential: $df = \dfrac{\partial f}{\partial u}du + \dfrac{\partial f}{\partial v}dv$

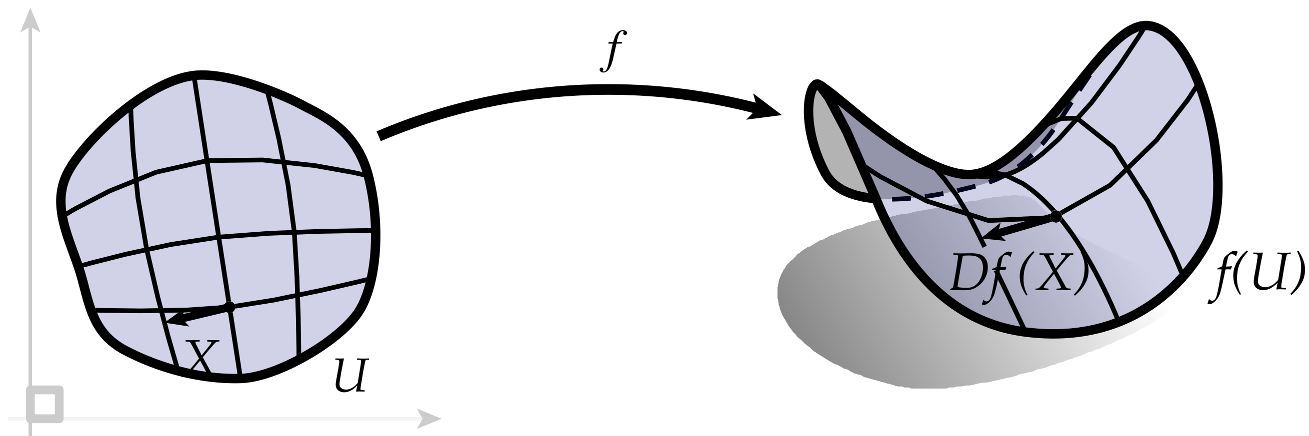
If point $p \in \mathbb{R}^2$ moves with velocity $X = [u, v]^T$ by ϵ, the movement of f_p is:

$$\Delta f_p \approx \frac{\partial f}{\partial u}(\epsilon u) + \frac{\partial f}{\partial v}(\epsilon v) = \epsilon \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right] \begin{bmatrix} u \\ v \end{bmatrix} = \epsilon [Df_p] X$$

$$Df_p := \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\right] \in \mathbb{R}^{3\times 2}$$

Df_p: differential (Jacobian), a linear map.

Differential of a Surface



Total differential: $df = \frac{\partial f}{\partial u} du + \frac{\partial f}{\partial v} dv$

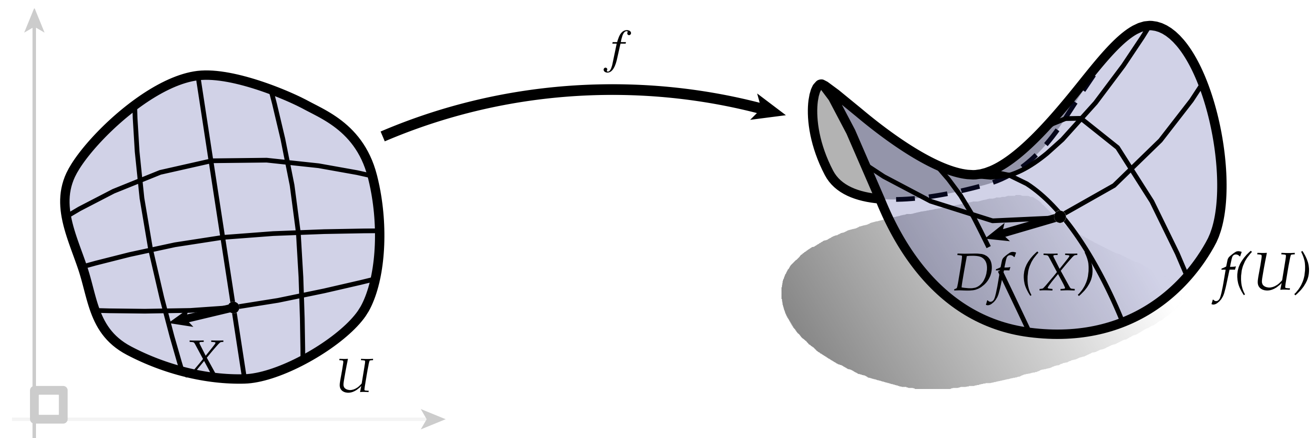
If point $p \in \mathbb{R}^2$ moves with velocity $X = [u, v]^T$ by ϵ, the movement of f_p is:

$$\Delta f_p \approx \frac{\partial f}{\partial u}(\epsilon u) + \frac{\partial f}{\partial v}(\epsilon v) = \epsilon \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v} \right] \begin{bmatrix} u \\ v \end{bmatrix} = \epsilon [Df_p] \boxed{X}$$

$$Df_p := \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v} \right] \in \mathbb{R}^{3 \times 2}$$

velocity in the 2D domain

Differential of a Surface



Total differential: $df = \frac{\partial f}{\partial u} du + \frac{\partial f}{\partial v} dv$

If point $p \in \mathbb{R}^2$ moves with velocity $X = [u, v]^T$ by ϵ, the movement of f_p is:

velocity in 3D space

$$\Delta f_p \approx \frac{\partial f}{\partial u}(\epsilon u) + \frac{\partial f}{\partial v}(\epsilon v) = \epsilon \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v} \right] \begin{bmatrix} u \\ v \end{bmatrix} = \epsilon \boxed{[Df_p] \boxed{X}}$$

$$Df_p := \left[\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v} \right] \in \mathbb{R}^{3 \times 2}$$

velocity in 2D domain