Humanoid Robotics

Manipulation 4: Grasping and Pushing

Maren Bennewitz

HRL

UNIVERSITÄT BONN

Goal of This Chapter

- Get an overview of common **robotic grasping tools**
- Learn about the **fundamentals of grasp planning** as well as **learning-based approaches**
- Get an overview of **object pushing** approaches

How To Manipulate?

- Robots **are not limited** to only one way of manipulating objects
- Objects can be **grasped**, **sucked**, or **pushed**
- **Which action** to use depends on the **type of object** (e.g., rigid, deformable, or delicate) and the **situation** (e.g., cluttered or free space)
- Different end-effectors enable robots to perform various types of object interaction

End-Effector Overview

- Parallel-jaw gripper
- Antipodal grasp
- Standard in robotics, many variants

Robotiq

OnRobot

Franka Robotics

End-Effector Overview

- Parallel-jaw gripper

[Zhu et al., RSS, 2022]

End-Effector Overview

- Multi-finger gripper
- Good for in-hand manipulation or not uniformly shaped objects (better enclosing)

Robotiq

SoftGripping

Alegro

End-Effector Overview

- Multi-finger gripper
- Good for in-hand manipulation or not uniformly shaped objects (better enclosing)

[Patel et al., ICRA, 2025]

End-Effector Overview

- Human-like 5-finger gripper
- Helps mimicking human dexterity

6 DOF
Psyonic

19 DOF
Shadow

1 DOF
QB-Robotics

End-Effector Overview

- Human-like 5-finger gripper
- Helps mimicking human dexterity

Courtesy: Psyonic

End-Effector Overview

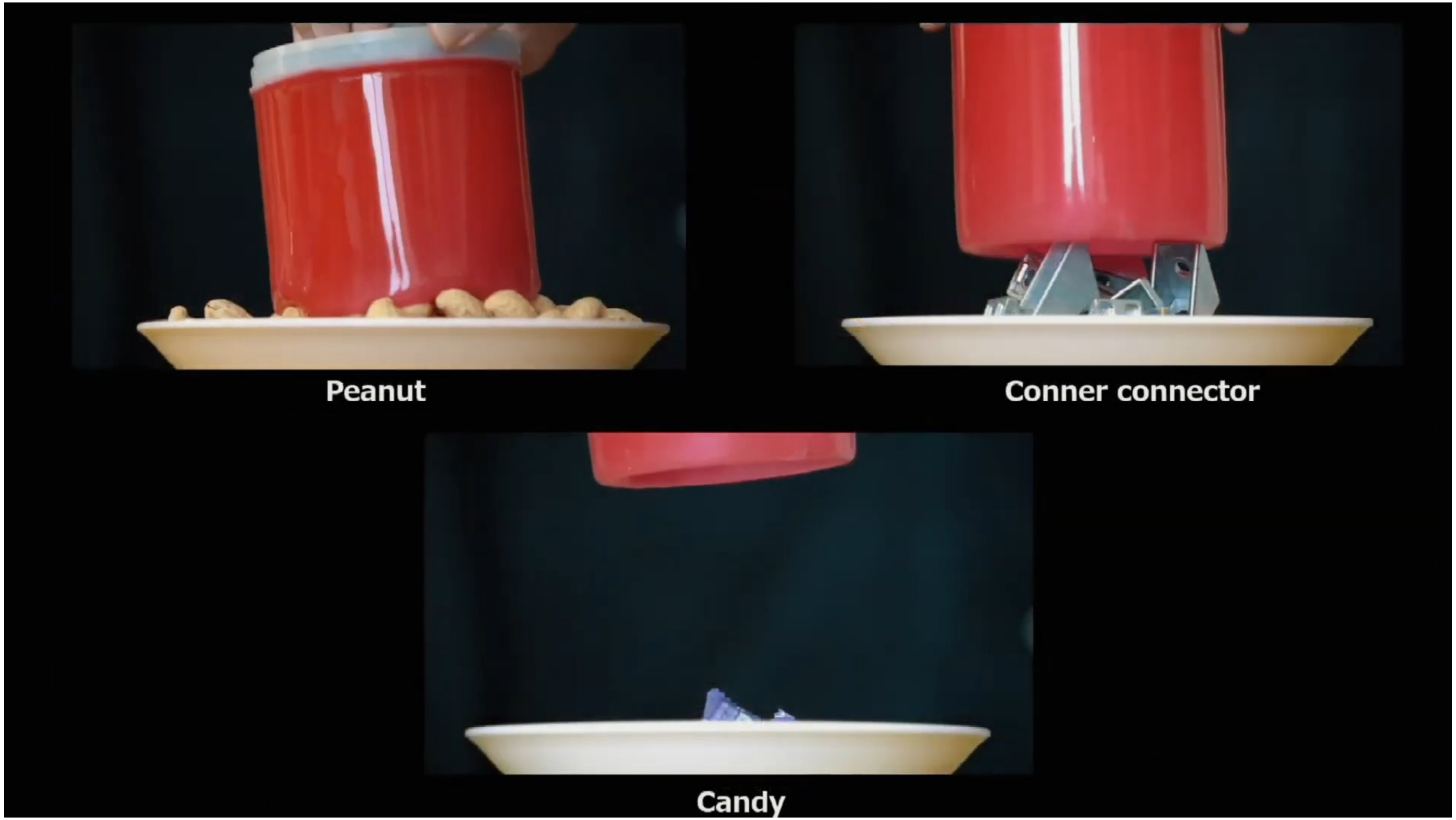
- Vacuum Gripper
- Simplifies “grasping”, enables grasping objects in high clutter or difficult shapes

OnRobot

https://www.youtube.com/watch?v=3OFPGEmf5IA

Unconventional End-Effectors

- ROSE-Gripper: **RO**tation-based **S**queezing gripp**E**r



[Bui et al., RSS, 2023]

Grasp Planning

- Obtaining **complete control** of an object's motion is a fundamental component of robotic manipulation
- **General idea:** Use end-effectors (fingers) to hold an object
- **Definition:** The application of forces at a set of contact points to restrain an object's motion

What Makes a Good Grasp in General?

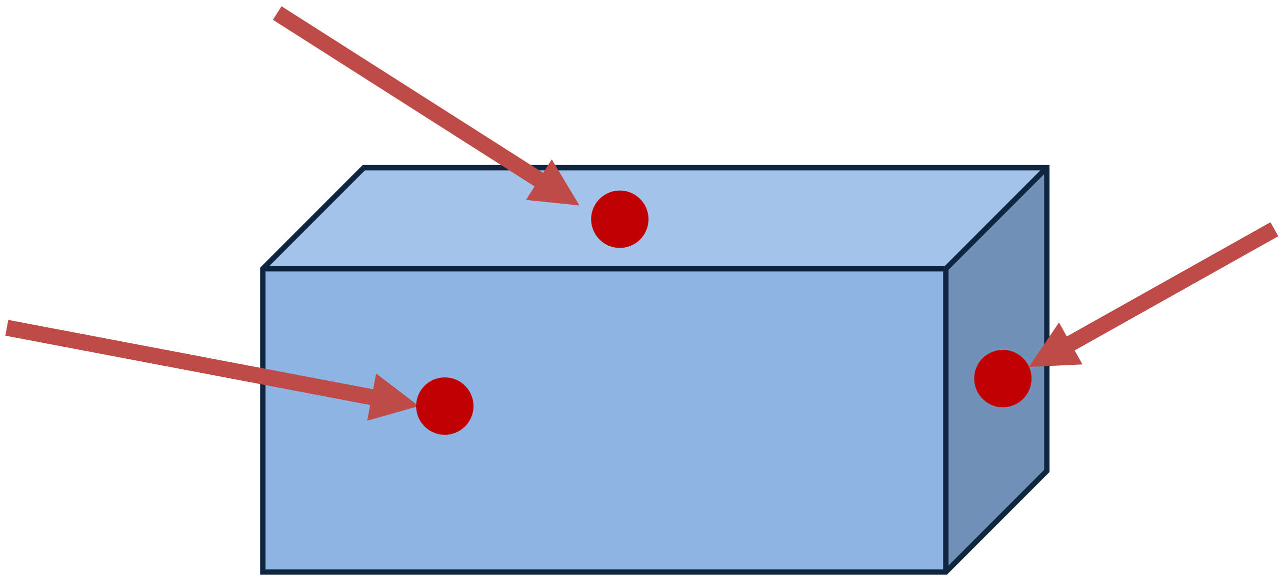
1. Lifts the object
2. Ensures minimal unexpected shifts of the object’s pose (slipping)
3. Keeps the object grasped during transfer
4. Avoids contact with nearby objects
5. Enables successful placing of the object
6. Within the robot’s reach (reachability map)

Why is Grasp Planning Hard?

- **High-dimensional** gripper configuration (fingers + wrist)
- **Contact-point selection**: Ideal contacts yield robust grasps, but feasible contacts are constrained by geometry
- **Robustness evaluation**: Assess grasps to ensure robust grasping under uncertainty or deviations
- **Relevant factors**: friction, object gravity, external forces
- **Collision avoidance**: During execution, the entire robot body including the object must avoid unintended collisions

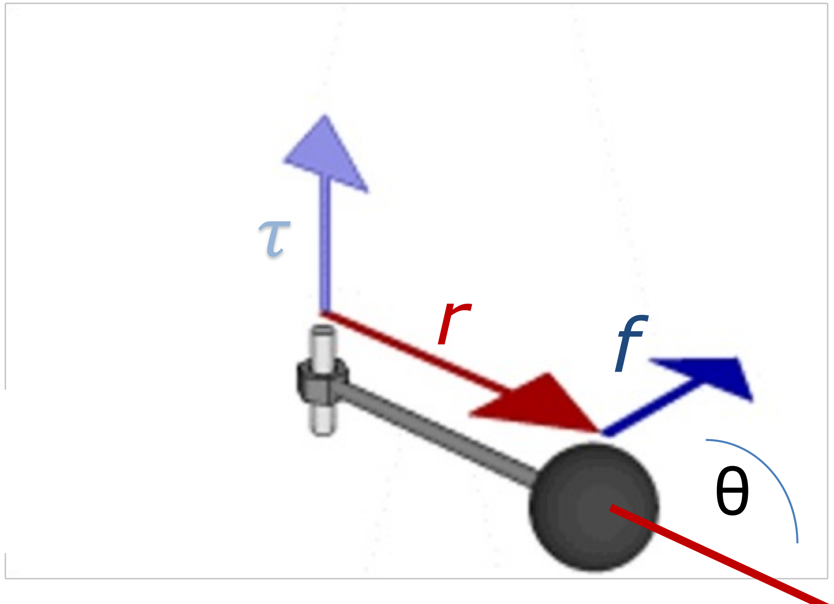
Grasp Modeling

- **Contact points** define the **quality of a grasp**
- **Point-on-plane contact models:**
  Commonly used for grasping since the possible contact points for most objects are almost always on surfaces



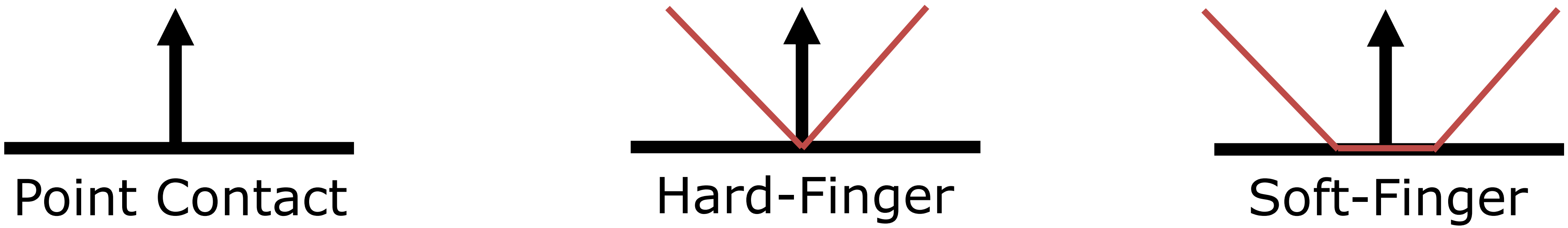
Basics: Force and Torque

- Newton's definition of force: $f = m * a$ with mass *m* and acceleration *a*
- Total force *f* = sum of forces on a rigid body
- Torque = rotational effect of a force
- Torque direction: $\tau = r \times f$, magnitude: $|\tau| = rf\sin(\theta)$
- Wrench $w = (f, \tau)$



Point-on-Plane Contact Models

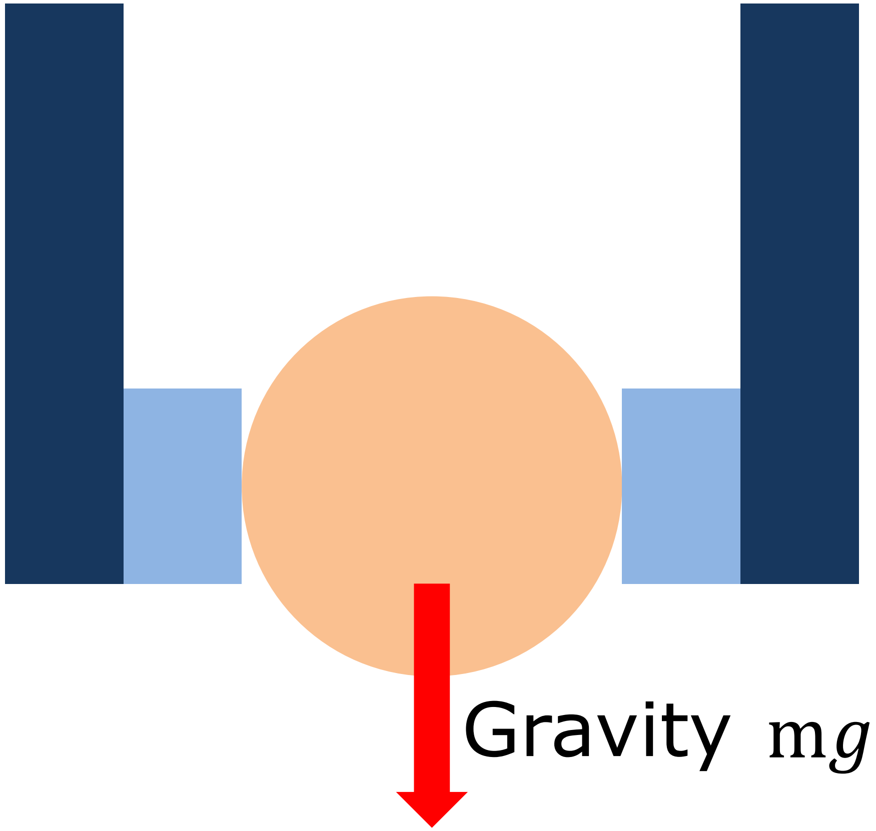
- **Frictionless point contact**:
  Forces can only be applied along the surface normal
- **Point contact with friction (hard-finger contact)**:
  Forces can be applied in directions other than just the surface normal, defines a **friction cone**
- **Soft-finger contact**:
  Allows for torque around the surface normal axis and includes a friction cone for the forces



Parallel-Jaw (2-Finger) Grasping in 2D

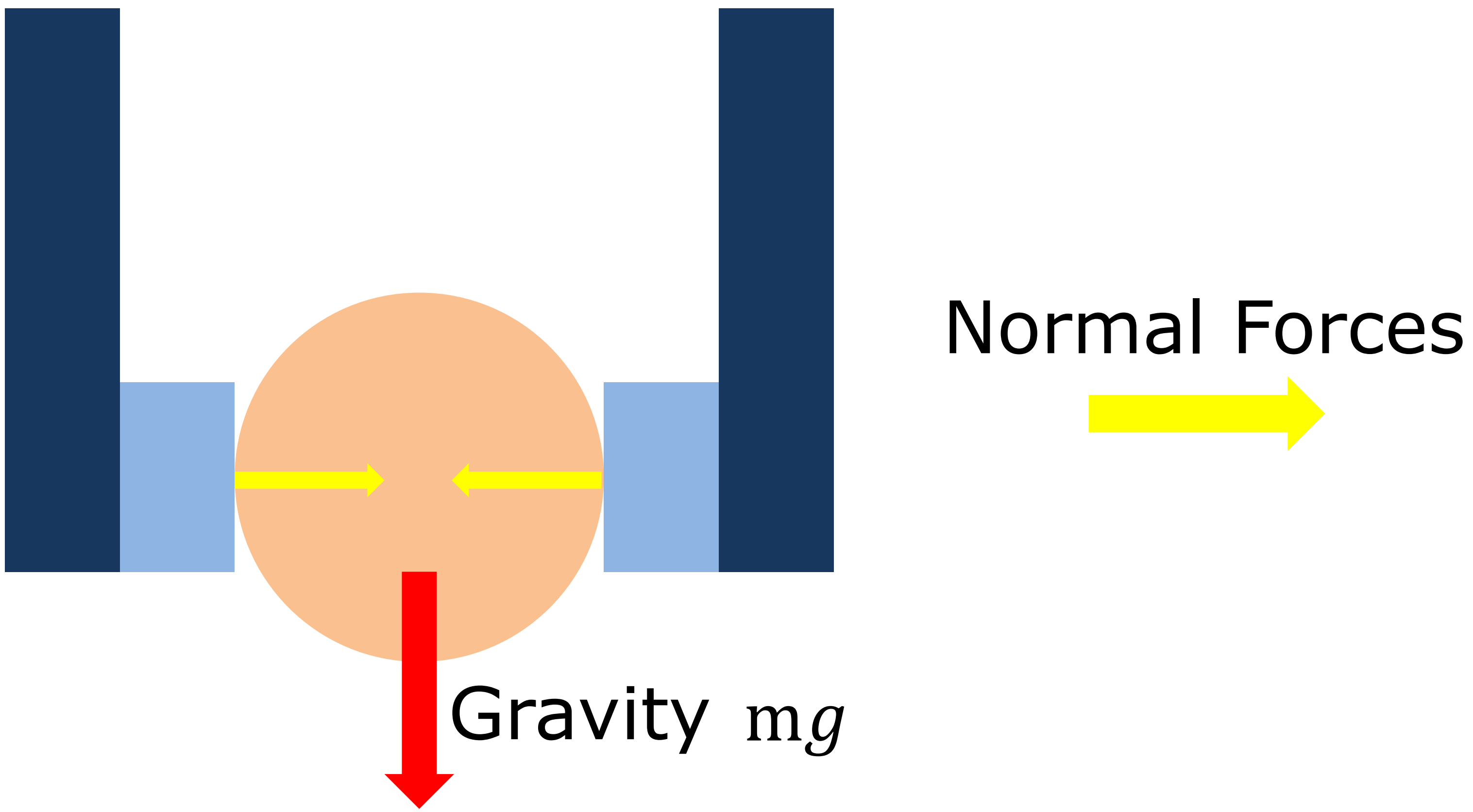
- How to **counter** gravity force to **lift an object**?

Robotiq



Parallel-Jaw (2-Finger) Grasping in 2D

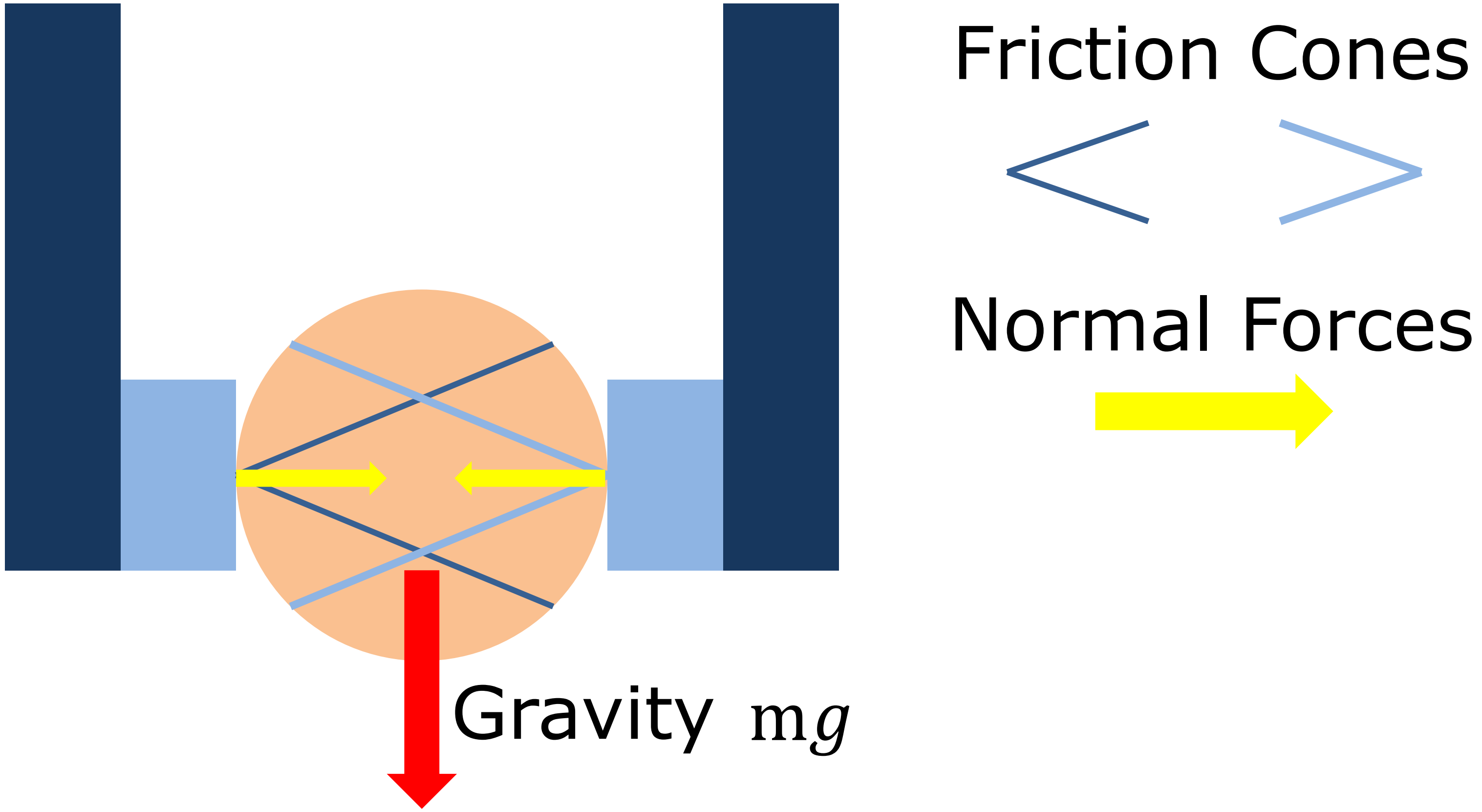
- How to **counter** gravity force to **lift an object**?



Parallel-Jaw (2-Finger) Grasping in 2D

- How to **counter** gravity force to **lift an object**?

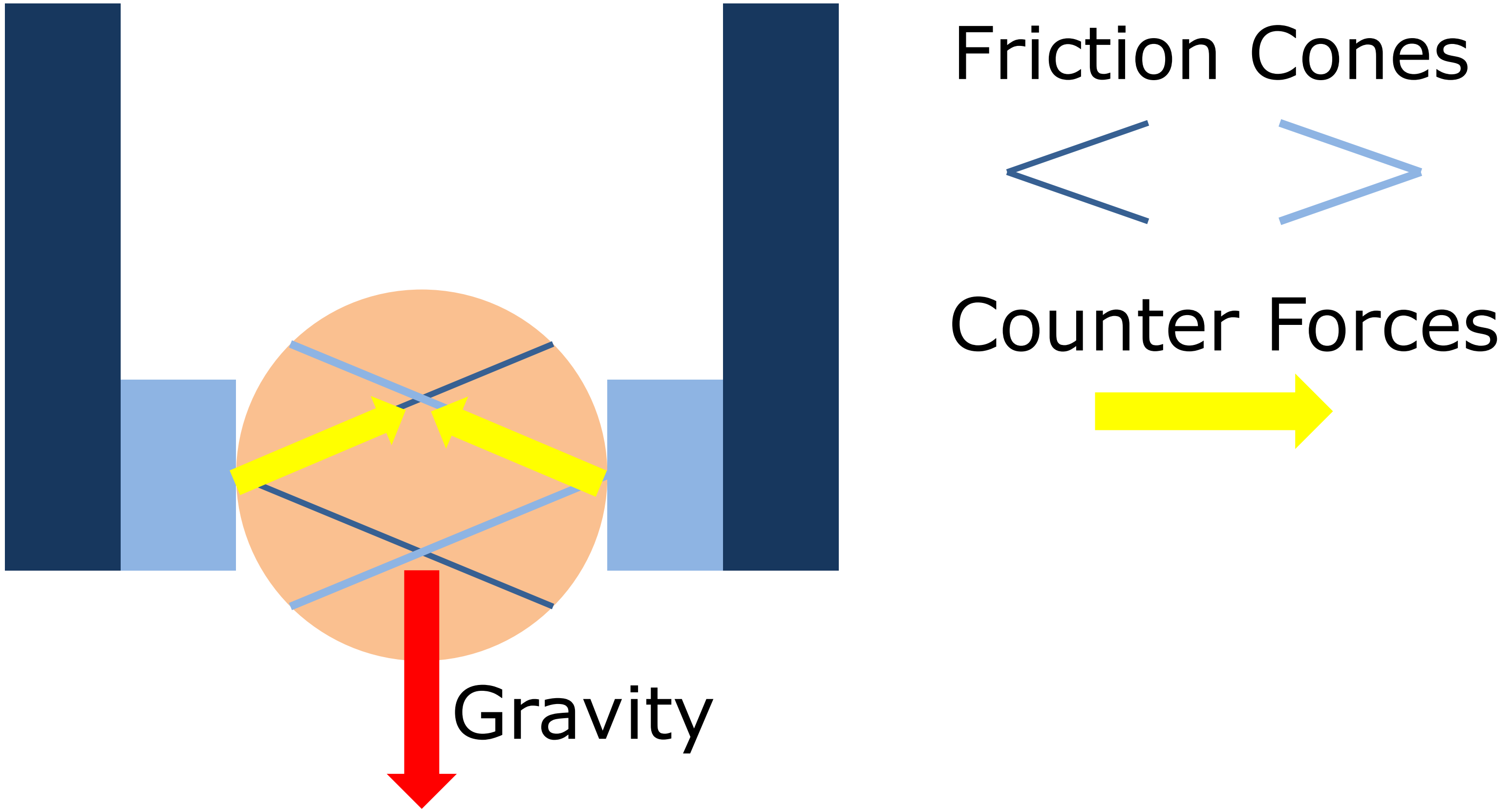
$$\| f_t \| \leq \overset{\text{friction coefficient}}{\mu} F_n$$



Parallel-Jaw (2-Finger) Grasping in 2D

- How to **counter** gravity force to **lift an object**?
- In the friction cone, there are forces that counteract the gravitational force

Force has to be greater than or equal to the gravitational force acting on the object

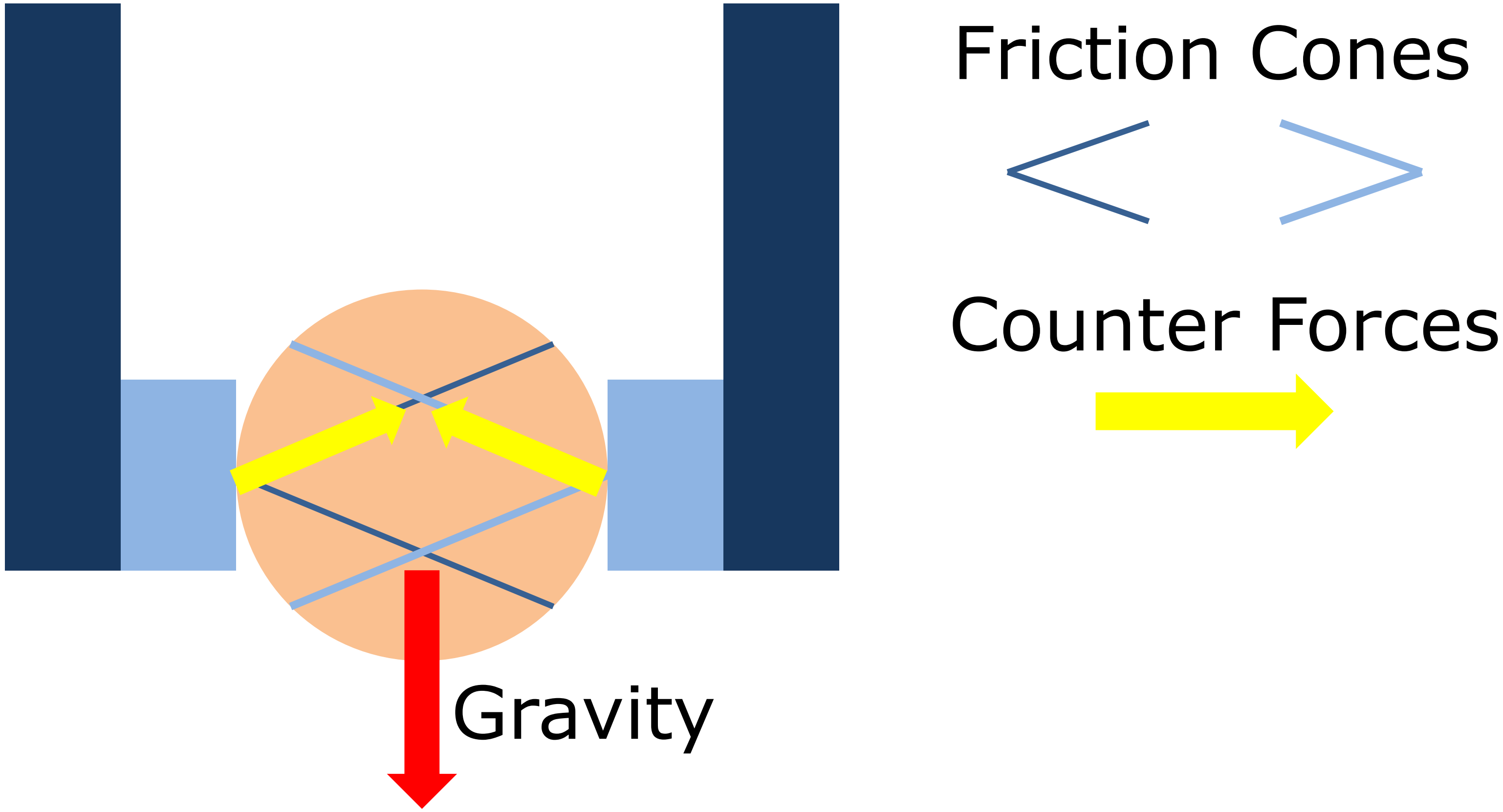


Parallel-Jaw (2-Finger) Grasping in 2D

- How to **counter** gravity force to **lift an object**?
- In the friction cone, there are forces that counteract the gravitational force

$$2\mu F_n \geq mg$$

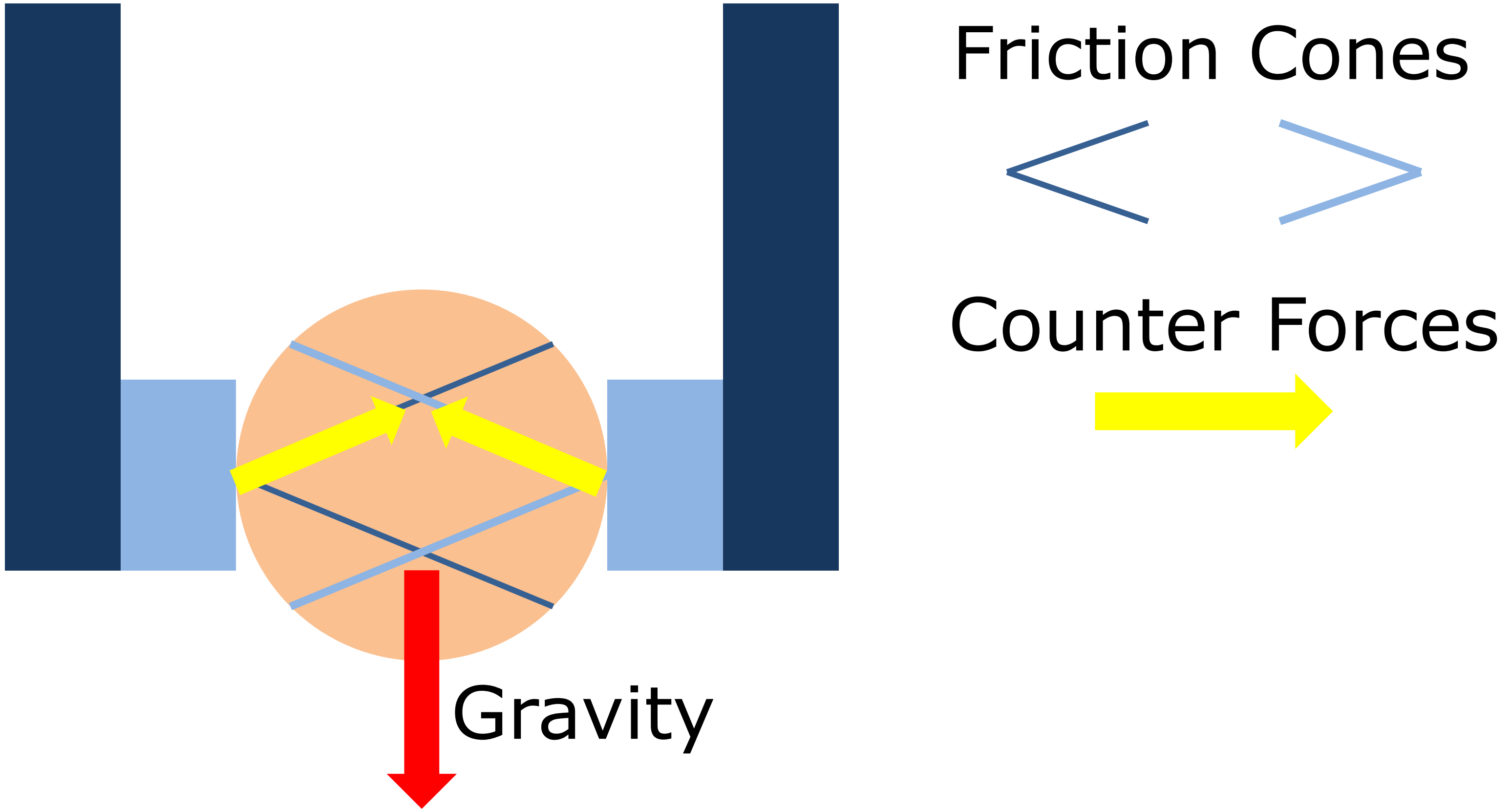
$$F_n \geq mg/2\mu$$



Parallel-Jaw (2-Finger) Grasping in 2D

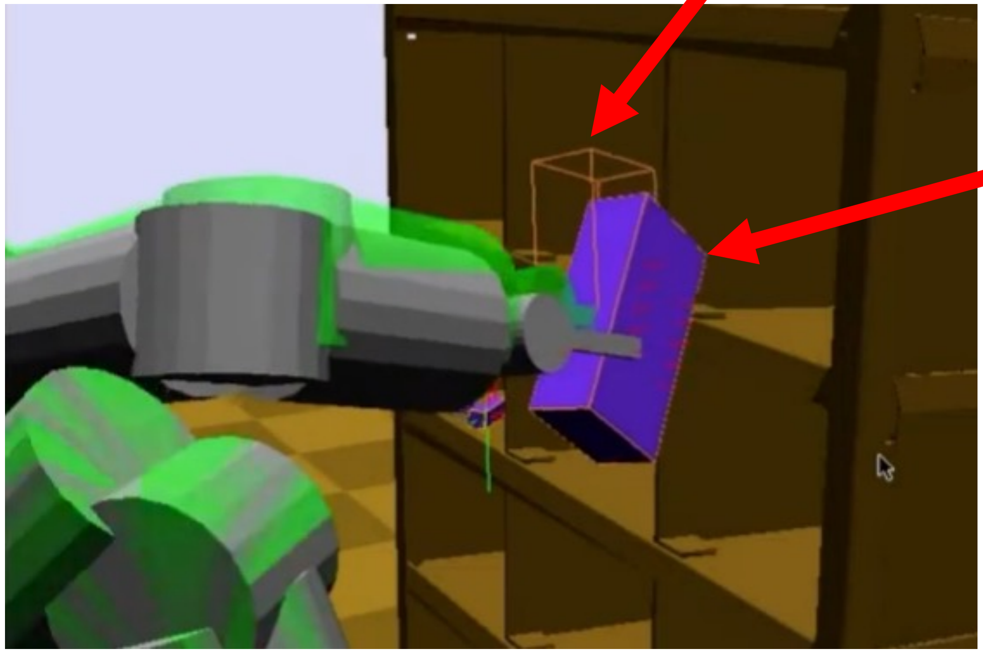
- How to **counter** gravity force to **lift an object**?
- In the friction cone, there are forces that counteract a given gravitational force

Given grip force F_n and friction μ, **maximum mass of load:** $2\mu F_n/g$



Parallel-Jaw (2-Finger) Grasping in 3D

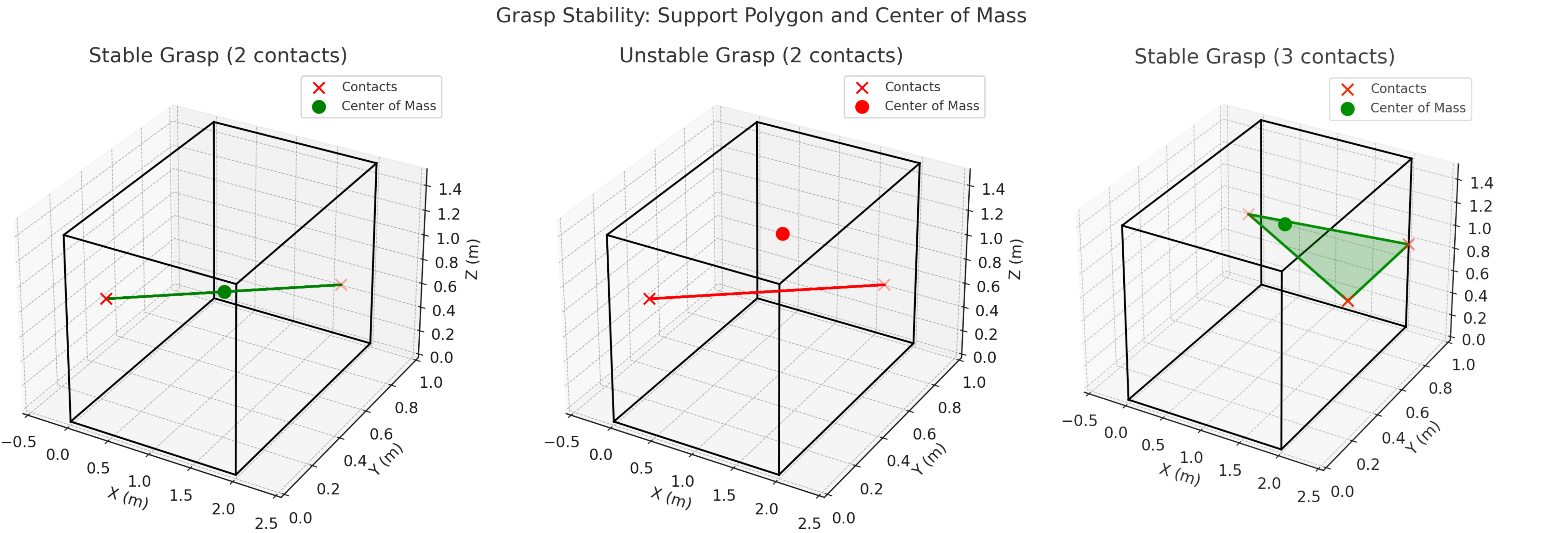
- With only two contact points, **unable to resist** torque about the axis
- This **may be also true** for more contact points



Courtesy: K. Hauser

Parallel-Jaw (2-Finger) Grasping in 3D

- One possible solution: **Consider the support polygon**

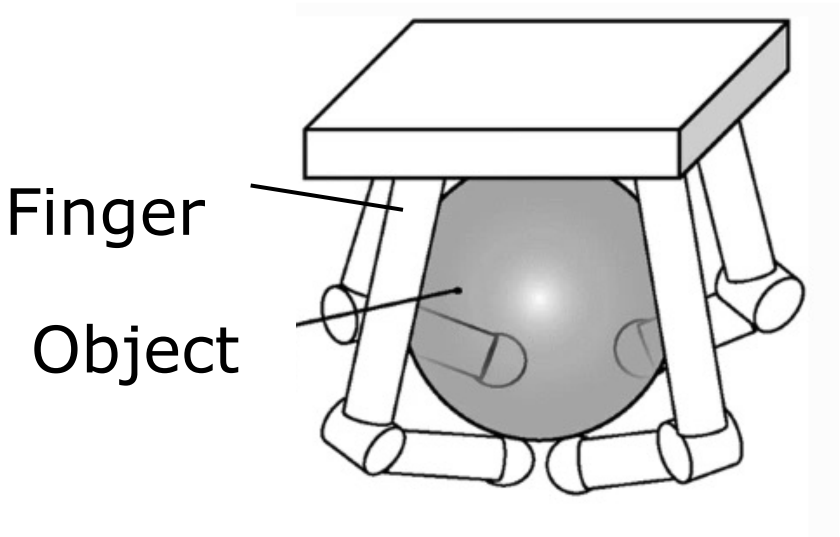


Grasp Quality Evaluation

- **Ideal grasps show closure**
- Grasp closure: Grasp can be **maintained for every possible disturbance load**
- Firm closure allows the robot to maintain its grasp, e.g., even if someone tries to hit it out of the hand
- Two common types of closures: **form** and **force closure**

Form Closure

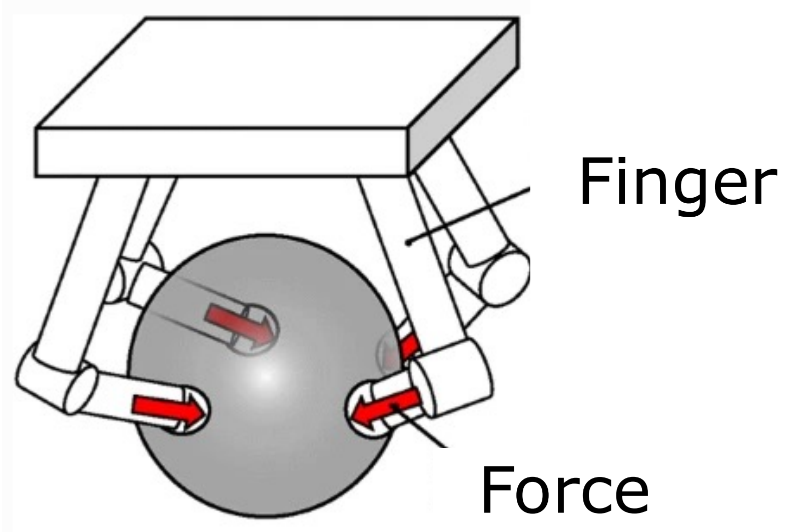
- Fixed grasp that **locks** the object between the fingers
- **Geometric constraints** alone prevent the object from moving
- No wrench (force + torque) can move the object
- Guaranteed immobilization
- But extremely sensitive to shape, requires many contacts
- Often not feasible to compute



Kim et al., ROBOMECH, 2019

Force Closure

- Applies **sufficient counter forces** at the contact points in order to **resist any external wrench**
- Relies on friction and generally requires **fewer contact points** than required for form closure
- However, there might **not be a solution** for every object and gripper



Kim et al., ROBOMECH, 2019

Form Closure vs. Force Closure

- **Form closure**:
  - Relies solely on the geometry of contact to lock the object
  - Friction-independent, but impractical for most robot hands due to the required precision of contact
- **Force closure**:
  - Uses forces at contacts to resist external disturbances
  - More widely applicable, but sensitive to friction and contact stability

Perception for Grasp Planning

- Real scenes: unknown, deformable, cluttered objects
- Analytical methods (force/form-closure) need precise models
- Perception-driven methods rely only on sensor data
  - Sampling-based
  - Learning-based

Sampling-Based Grasp Planning

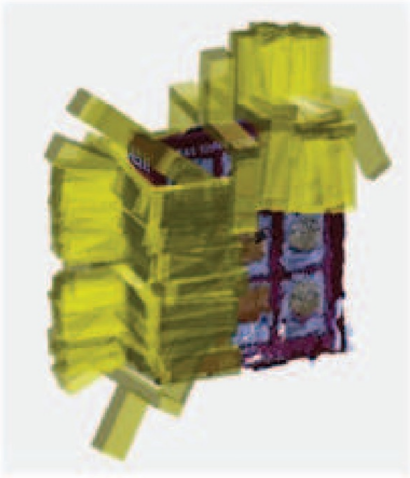
- Sample and evaluate **contact candidates** directly on object surface or RGBD
- Use **predefined hand models** that come into contact when fingers close
- **Good grasps**: Combination of stable contact points **and** feasible hand configuration
  - Contact points **ignoring the actual hand geometry**, can lead to contact locations **unreachable** for the hand
  - **Feasible hand configuration** can generate a weak grasp in the presence of small perturbations

Learning-Based Grasp Planning

- **Sample & score:** point clouds, analytic labels (GPD)
- **Pixel regression:** RGB-D images, per-pixel labels (AnyGrasp)
- **Volumetric CNN:** TSDF volumes, voxel-wise labels (VGN)

Grasp Pose Detection (GPD)

- **Input:** Raw point cloud of the scene
- Approach:
  - **Sample** pairs of surface points with estimated grasp pose
  - **Encode** local geometry around grasp into heightmap descriptors
  - **Score** candidates with network
- **Output:** Ranked 6-DoF grasp poses
- Relies on **analytic force-closure labels during training** on sampled candidates



Ten Pas et al., "Grasp Pose Detection in Point Clouds", IJRR 2018

GPD Demonstration

Ten Pas et al., “Grasp Pose Detection in Point Clouds”, IJRR 2018

AnyGrasp

- Dense **per-pixel** 7-DoF grasp prediction from a single RGB-D view (3D point is assumed center point of grasp)
- **Input:** single-view RGB-D image
- **Output:** dense per-pixel 7-DoF grasp candidates $(x, y, z, \alpha, \beta, \gamma, width)$ and quality score
- Evaluates grasps at each pixel in **one pass**
- Relies on **dense per-pixel quality labels** during training

https://graspnet.net/anygrasp.html

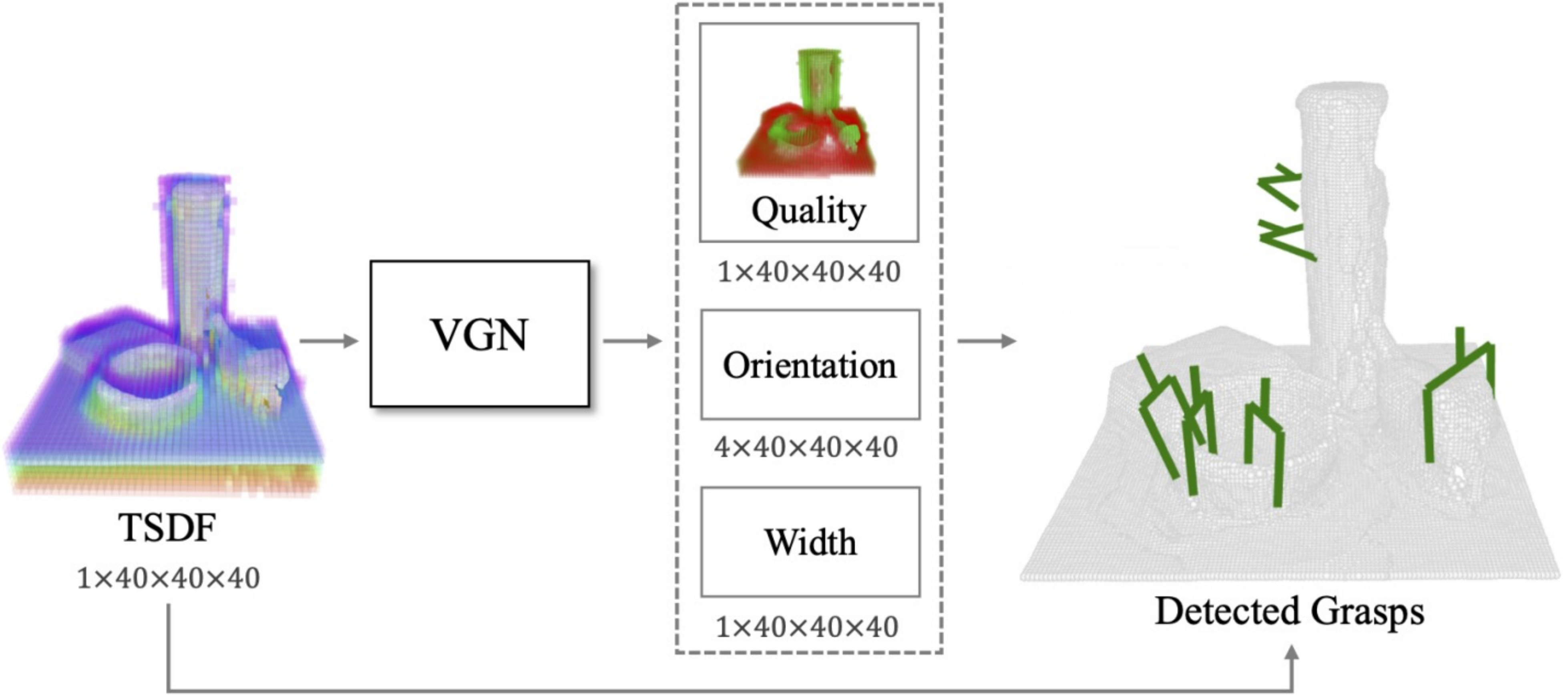
Fang et al., "AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains", TRO 2023

Volumetric Grasping Network (VGN)

- Real-time **volumetric** grasp prediction via TSDF fusion
- **Input**: fused TSDF voxel grid
- **Output:** per-voxel 7-DoF grasp candidates $(x, y, z, \alpha, \beta, \gamma, width)$ and quality score
- Evaluates grasps at each voxel in **one pass**
- Relies on **sparse voxel-wise ground truth labels** for training from simulated grasp trials

Breyer et al., "Volumetric Grasping Network: Real-time 6-DoF Grasp Detection in Clutter", CoRL 2020

Volumetric Grasping Network (VGN)



Breyer et al., "Volumetric Grasping Network: Real-time 6-DoF Grasp Detection in Clutter", CoRL 2020

GPD vs. AnyGrasp vs. VGN

	GPD (point cloud)	AnyGrasp (RGBD image)	VGN (fused TSDF voxel grid)
Runtime	≈0.5–1 s per cloud	≈100 ms per frame	≈10 ms per volume
Success Rate	~93 % on novel objects	93.3 % bin-clear on 300 unseen objects	~92 % clearance in clutter
Temporal Tracking	No (single snapshot)	Yes (frame-to-frame tracking)	No (requires re-fusion per frame)
Closed-Loop Suitable	No	Yes	Yes
Collision Checking	explicit during select	implicit via depth/collision test	Uses 3D scene to directly learn collision-free grasps
Approach Direction Filtering	Yes (axis constraints)	No	No
Object-Selective	Yes (specify point indices)	Yes (pixel-region masks)	No
Pros	• Interpretable	• Full 7-DoF • Dynamic tracking	• Real time, full 7-DoF • Leverages full 3D
Cons	• Slow • Static only	• Black box	• Needs TSDF fusion

Real-Time Grasp Correction with Perception

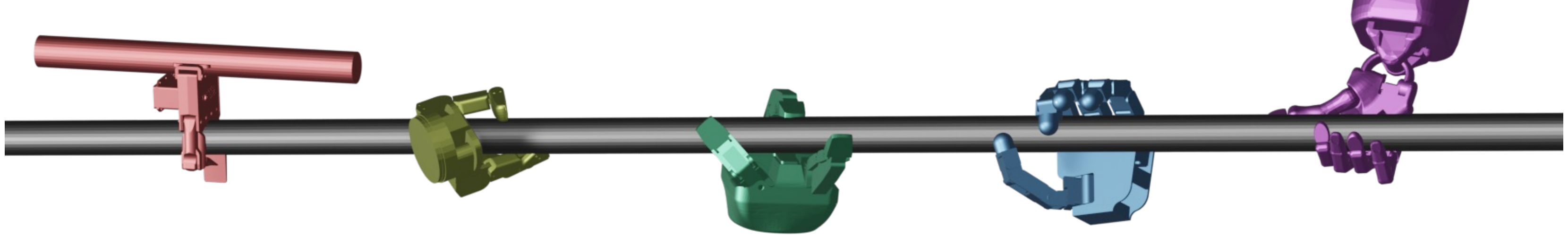
- Open loop: Plan once and go blind
- Closed loop: **Continuously update** grasp pose during approach **to correct** for errors and disturbances
- **Sense:** Evaluate point cloud or RGB-D at each control cycle, fuse into TSDF if necessary
- **Re-compute grasp:** Re-evaluate grasp hypotheses for current RGB-D (AnyGrasp) or TSDF volume (VGN)
- **Adjust motion:** Send incremental velocity commands to align gripper to updated pose

From Parallel-Jaw to Multi-Finger Grasping

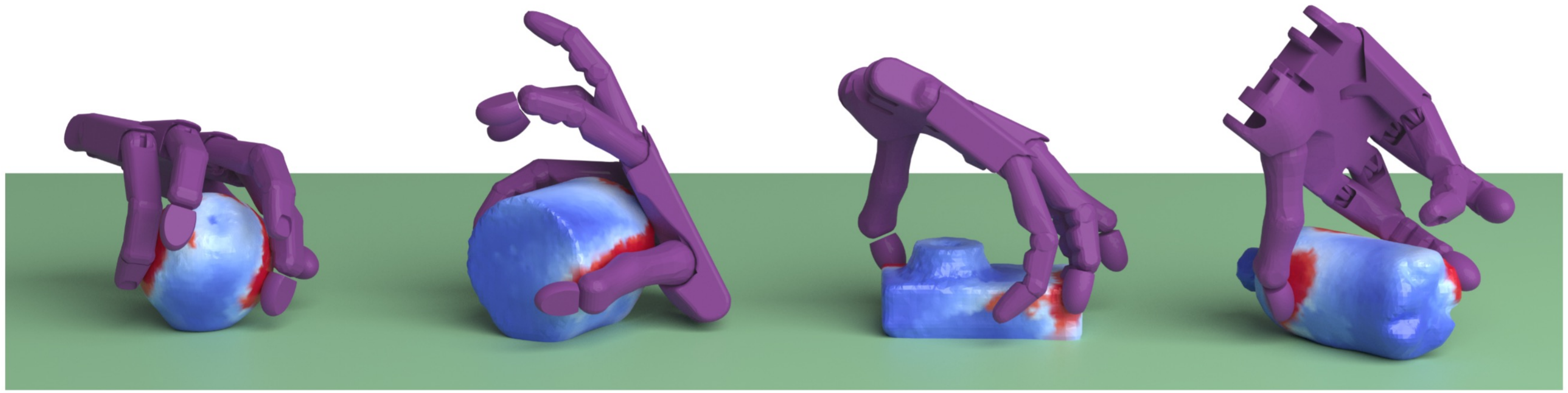
- GPD, AnyGrasp, VGN: 2-finger grasp prediction
- However, humanoids typically have multi-finger hands
- 3–5 fingertips leads to **combinatorial increase** in candidate poses
- Coordination of finger trajectories and contact sequence

[Winkelbauer et al., IROS 2022]

GenDexGrasp

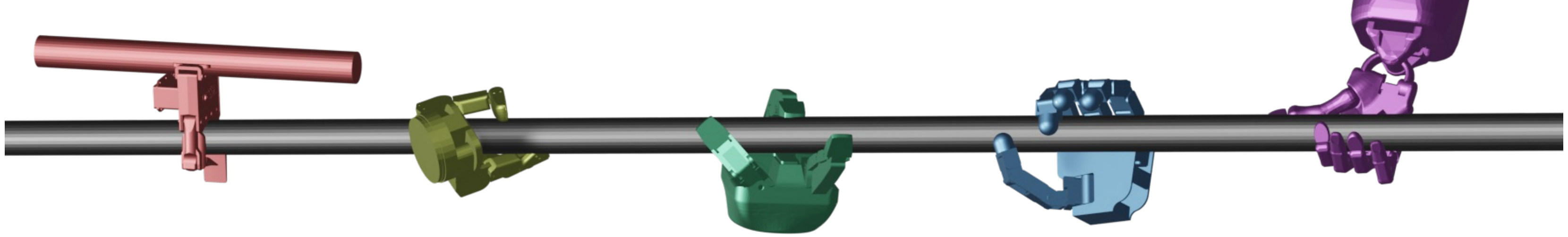


- **Learns contact maps** where objects should be grasped
- **Independent** of a specific robot hand
- Propose a dataset capturing **force closure** for different objects and hands
- **Predicts contact points** for an unseen object and **optimizes** the robots **hand pose** to realize them

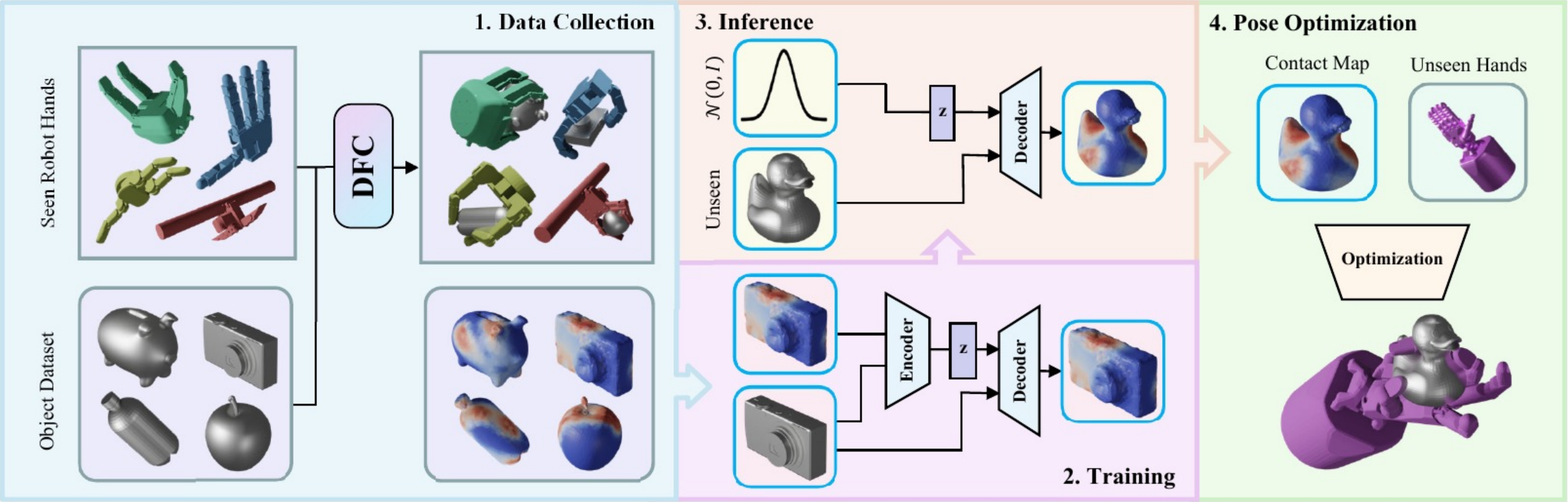


Li et al., "GenDexGrasp: Generalizable Dexterous Grasping", ICRA 2023

GenDexGrasp



- **Train a CVAE** with original contact maps and objects
- At inference, sample a latent vector z to decode a new contact map **conditioned on the object**
- **Optimize the hand pose** to match the contact map



Li et al., "GenDexGrasp: Generalizable Dexterous Grasping", ICRA 2023

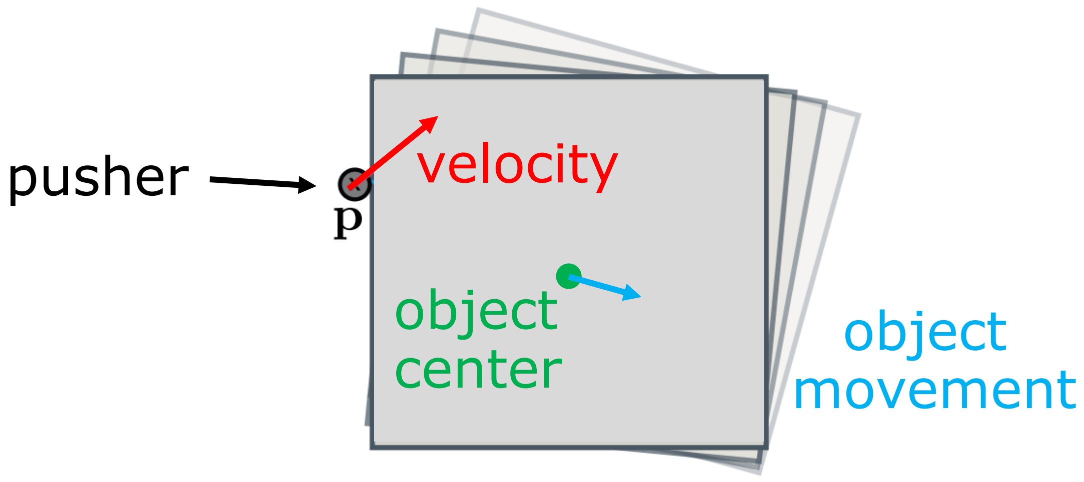
Pushing

Planar Pushing

- **Task:** Control the pose of an object in 2D using only "pushing" contacts
- **Interaction** between the **object and surface** must also be modeled
- Physics-based contact models use **friction** and **force** laws to **predict object motion during contact**

Analytical Model for Planar Pushing

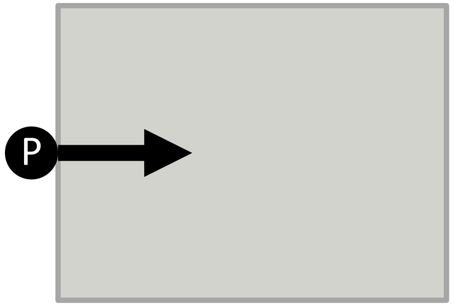
- **Only approximate** and far from perfectly modeling the process of planar pushing
- **Predicts the object movement** given the pusher velocity, the contact point as well as mass, force, and friction-related parameters



[Kloss et al., IJRR 2020]

Analytical Model for Planar Pushing

- Predicts the effect of a push
  - Is the push stable ("sticking contact")
  - Or will the pusher slide on object ("sliding contact")?
- **Sticking:** Velocity of the object at the contact point will be the **same as the velocity of the pusher**
- **Sliding:** Movement of pusher can be almost orthogonal to the resulting motion at the contact point

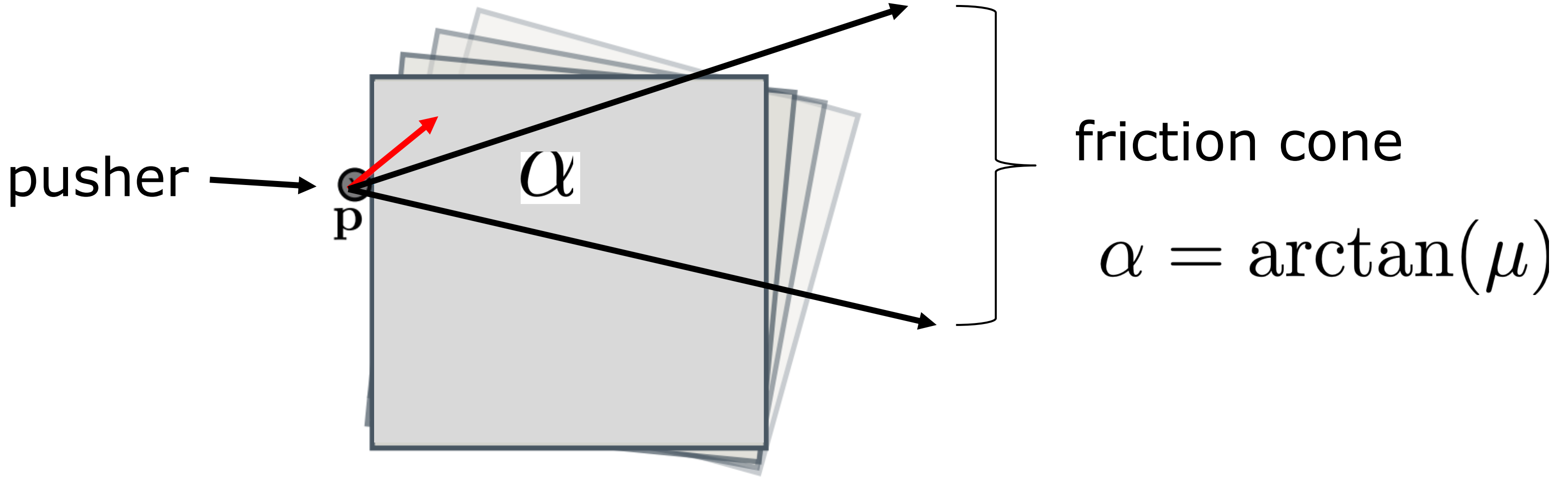


Sticking

Sliding

Analytical Model for Planar Pushing

- Find the left and right boundaries of the **friction cone**, i.e., the **forces** for which the **pusher is sticking**
- Calculated using the **opening angle of the friction cone**, which is defined by the friction coefficient between the pusher and object



[Kloss et al., IJRR 2020]

Analytical Model for Planar Pushing

- **Motion cone**: Set of all object motions that **maintain sticking contact** between the pusher and the object
- If the push velocity is outside of the motion cone, the **contact will slide** but the object still moves to some extend
- Otherwise, the contact is sticking and the **pusher velocity** is the effective **object velocity** at the contact point

Analytical Planar Pushing: Discussion

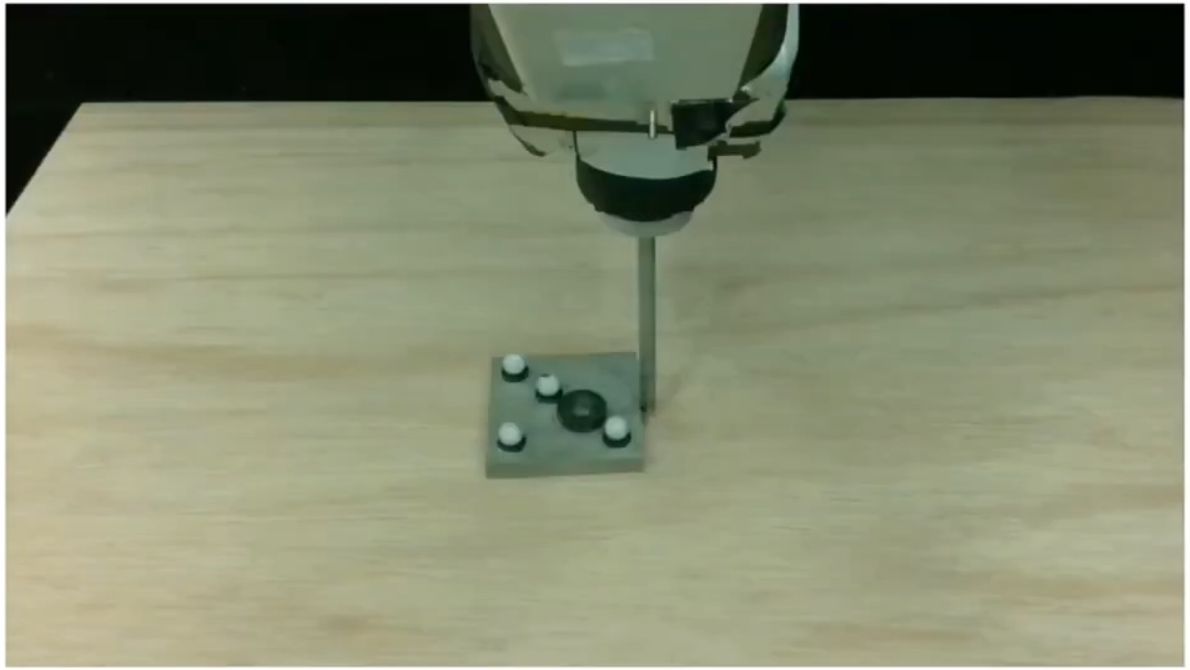
- Analytical model uses three **simplifying assumptions**:
  1. The force applied to the object is **high enough** to move the object, but **not to accelerate it**
  2. The pressure distribution of the object on the surface is **uniform**
  3. The friction between surface and object is **constant**
- Typically, Assumptions 2 and 3 are **violated in the real world**
- Furthermore, complex heuristics needed to handle **contact mode** switches

Learning to Push

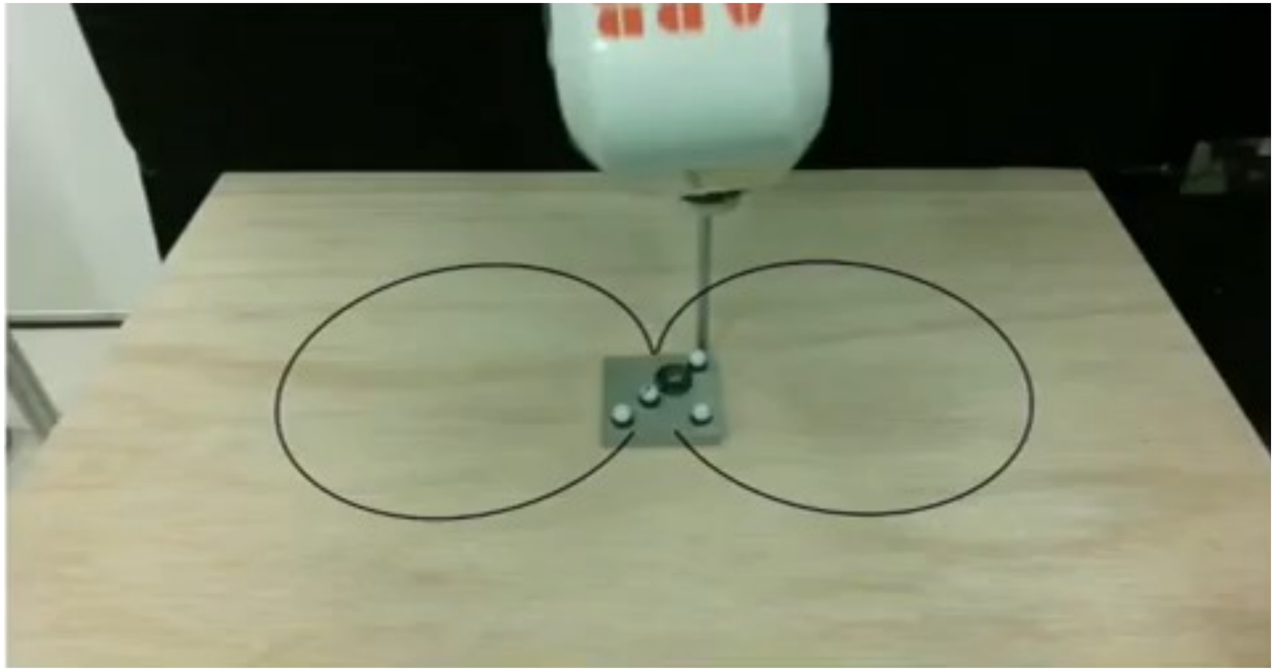
- Recent approaches do not rely on solely on analytical models anymore
- Different aspects of pushing can be learned (e.g., execution, world representation, touch sensing, etc.)
- Here: focus on learning the push execution, i.e., the action generation

Dynamical Model Learned from Data

Data collection



Execution



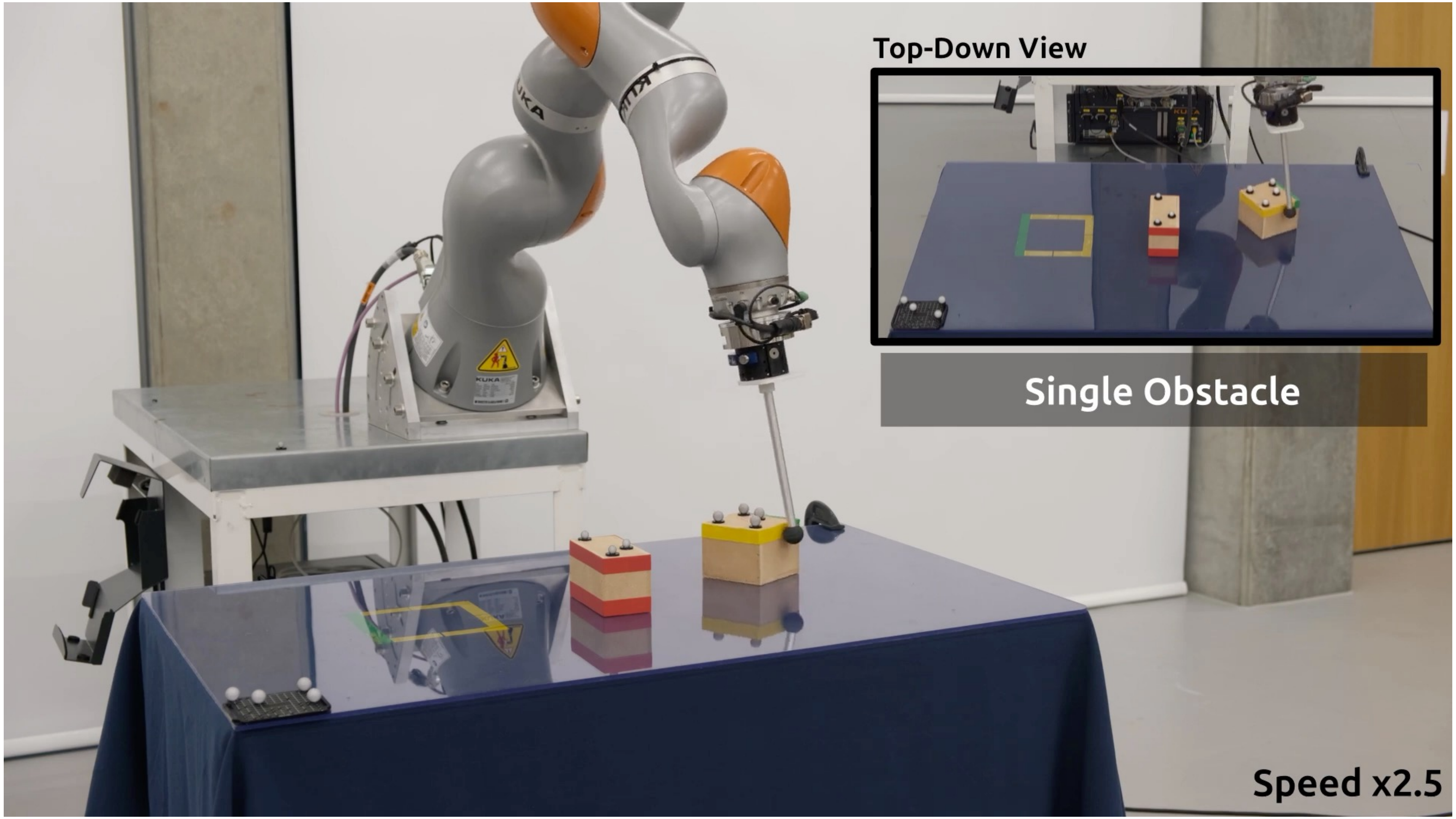
[Bauza et al., CORL 2019]

Learning Push Actions in Cluttered Scenes

- **Action**: push velocity in *x* and *y* direction for the next step
- **Observation**:
  - 2D occupancy grid map of environment
  - Current and target object pose
  - Pusher pose
- **Reward** based on:
  - Euclidean and angular distance between current and target pose
  - Collisions with other objects

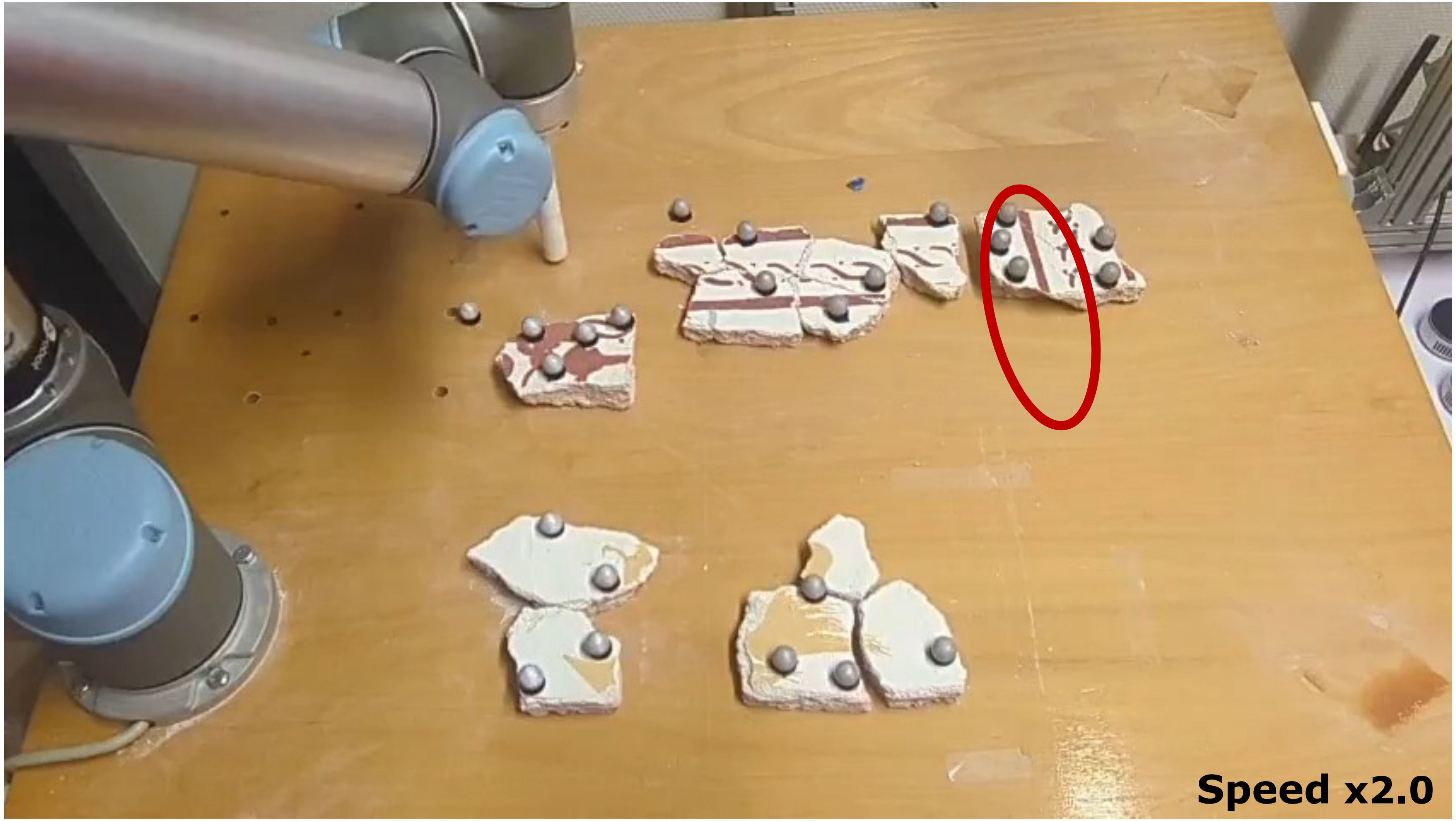
Dengler et al., Learning Goal-Directed Object Pushing in Cluttered Scenes with Location-Based Attention, IROS 2025

Learned Object Pushing



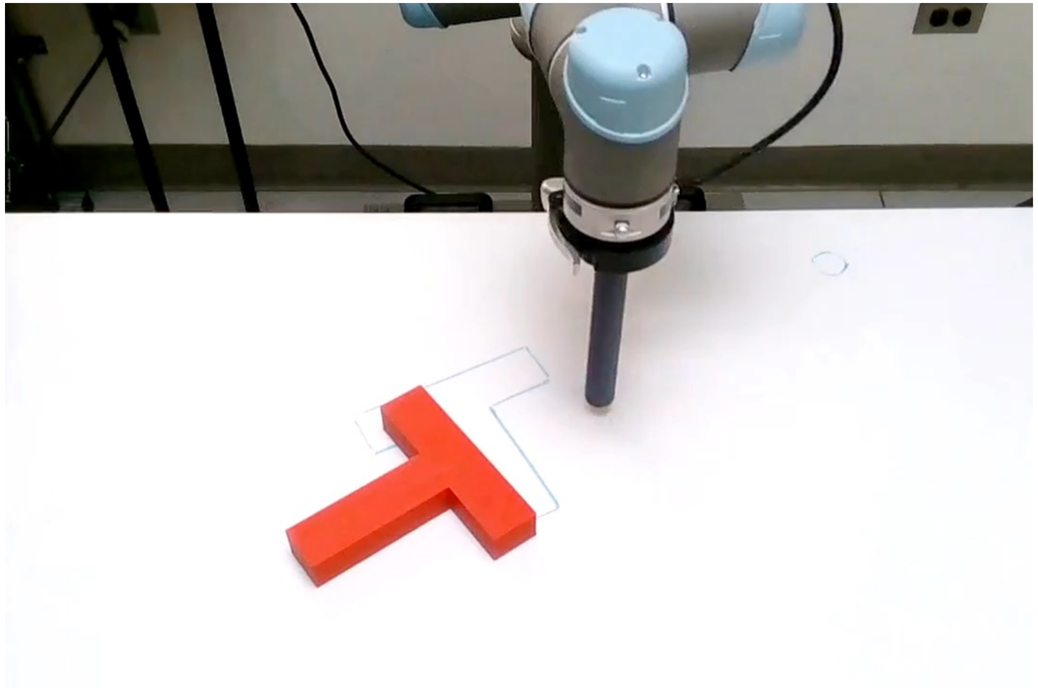
Dengler et al., Learning Goal-Directed Object Pushing in Cluttered Scenes with Location-Based Attention, IROS 2025

Generalization: Fresco Pushing



Diffusion Policies for Pushing

- Learns multiple valid pushing trajectories instead of a single solution from demonstrations
- Can handle uncertainty and contact-rich interactions
- Models a **distribution over action sequences** using a denoising process



https://diffusion-policy.cs.columbia.edu/

Learning-Based vs. Model-Based Pushing Approaches

Model-Based

- Uses explicit physics models (friction, dynamics, contact)
- Requires known object/ environment properties
- Provides interpretable predictions and guarantees
- Can struggle with uncertainty

Learning-Based

- Learns action-outcome relationships from data
- Handles unknown dynamics and real-world variability
- Improves through experience and trial-and-error
- Can generalize to new objects

Summary (1)

- **End-effector diversity**: From 2-finger to multi-finger and vacuum grippers
- **Grasp planning**: involves complex kinematics, robust contact selection, collision avoidance, and iterative re-planning under uncertainty
- Leverages analytical **force-** and **form-closure** calculations, **sampling-based planning**, large labeled **datasets**, and **deep learning** for predicting grasp success

Summary (2)

- **Planar pushing** controls 2D object pose through surface contacts
- Purely **analytical models** often break down when real-world conditions violate assumptions
- Learning-based pushing:
  - **Data-driven learning** of dynamical model
  - **Reinforcement learning** with occupancy grid for obstacles avoidance
  - **Diffusion policies** iteratively denoise trajectory distributions to handle multimodality and contact uncertainty

Literature (1)

- *Springer Handbook of Robotics,* Siciliano, Khatib, and Kröger, Springer, 2008
  https://link.springer.com/book/10.1007/978-3-540-30301-5
- *Fundamentals of Grasping, Springer Handbook of Robotics,* Prattichizzo, and Trinkle,
  https://web.stanford.edu/class/cs237b/pdfs/lecture/cs237b_lecture_7.pdf
- *Robotic Manipulation: Perception, Planning, and Control*
  Russ Tedrake, Course Notes for MIT 6.421, 2024,
  http://manipulation.mit.edu
- *Sample Efficient Grasp Learning Using Equivariant Models*
  Zhu, Wang, Biza, Su, Walters, Platt, Robotics: Science and Systems (RSS), 2022
- *Not Twisting Your Arm: Combining Grasping and Rotation in a Single Robot Hand Mechanism*
  Patel and Dollar, ICRA 2025 Workshop Handy Moves: Dexterity in Multi-Fingered Hands
- *Rose: Rotation-based squeezing robotic gripper toward universal handling of objects*
  Bui, Kawano, and Ho, Robotics: Science and Systems (RSS), 2019
- *Touch Feedback and Contact Reflexes Using the Psyonic Ability Hand*
  Akhtar, Cornman, Austin, and Bala, Myoelectric Controls Symposium, 2020

Literature (2)

- *Graspability Map: A Tool for Evaluating Grasp Capabilities*
  Roa, Dollar, Hertkorn, Borst, and Hirzinger,
  IEEE/RSJ Int Conf. on Intelligent Robots and Systems (IROS), 2011
- *A Two-Stage Learning Architecture that Generates High-Quality Grasps for a Multi-Fingered Hand*
  Winkelbauer, Bäuml, Humt, Thuerey, and Triebel,
  IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), 2022

Literature (3)

- *Combining Learned and Analytical Models for Predicting Action Effects from Sensory Data*
  Kloss, Schaal, and Bohg, The International Journal of Robotics Research (IJRR), 2022
- *A Data-Efficient Approach to Precise and Controlled Pushing*
  Bauza, Hogan, and Rodriguez, Conference on Robot Learning (CoRL), 2018
- *Learning Goal-Directed Object Pushing in Cluttered Scenes With Location-Based Attention*
  Dengler, Del Aguila Ferrandis, Moura, Vijayakumar, and Bennewitz, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2025
- *Diffusion Policy: Visuomotor Policy Learning via Action Diffusion*
  Chi, Xu, Feng, Cousineau, Du, Burchfiel, Tedrake, Song, The International Journal of Robotics Research (IJRR), 2023
- *github.com/mbreuss/diffusion-literature-for-robotics*
  Summary of key papers and blogs about diffusion models to learn about the topic