Object Manipulation: Grasping and Pushing

Maren Bennewitz, Nils Dengler, Rohit Menon
Humanoid Robots Lab, University of Bonn

Goal of This Chapter

- Get an overview of common **robotic grasping tools**
- Learn the **fundamentals of grasp planning**
- Understand different approaches to **object pushing**

How To Manipulate?

- Robots **are not limited** to only one way of manipulating objects
- Objects can be **grasped**, **sucked**, or **pushed**
- The decision 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

Different Types of End-Effectors

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

Robotiq

OnRobot

Franka Robotics

Different Types of End-Effectors

- Parallel-jaw gripper

Zhu et al., RSS, 2022

End-Effector Overview

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

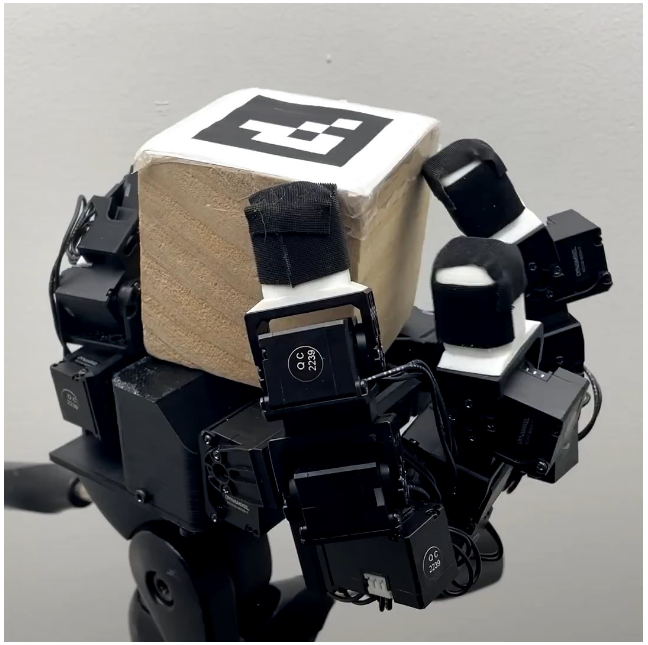
Robotiq

SoftGripping

Alegro

End-Effector Overview

- Multi-finger gripper
- Good for in-hand manipulation or not uniformly shaped obstacles (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

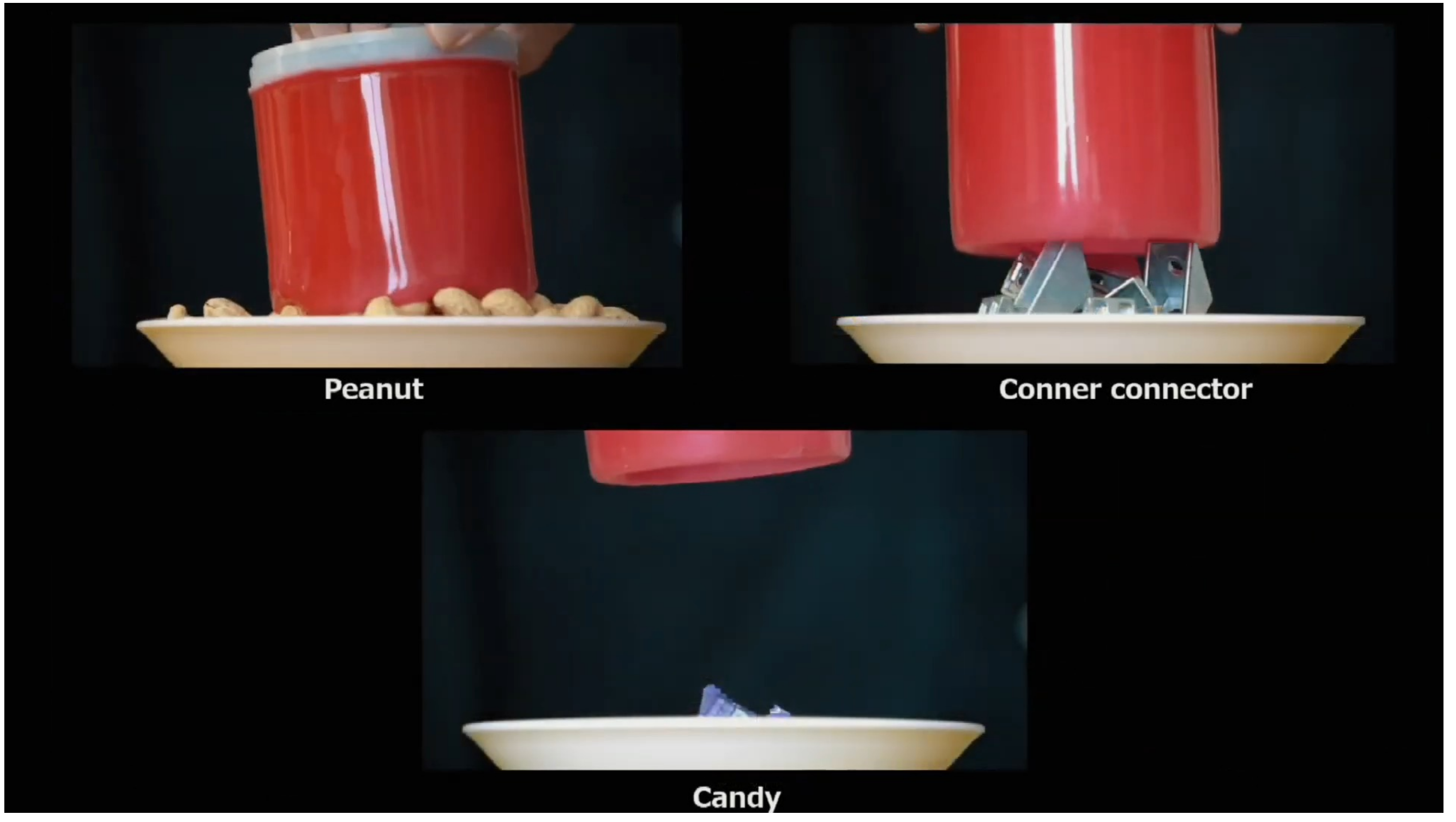
- Vacuum Gripper
- Simplifies grasping, enables grasping objects in high clutter or difficult shapes

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

- **Fundamental component** of robotic manipulation is obtaining **complete control** of an object's motion
- **General idea:** Using end-effectors (fingers) to hold an object relative to the hand
- **Definition:** The application of forces at a set of contact points to restrain an object's motion

Grasp Planning

- **Where to grasp** an object in order to perform a particular task?
- In the context of the video: Where to place my contacts to **immobilize** the object while being able to spray
- **Grasp analysis:** Given the contact points on an object, how stable is the resulting grasp (i.e., can the object slip?)

Courtesy: Psyonic

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. Leads minimal 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 (hand + 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 must avoid unintended collisions

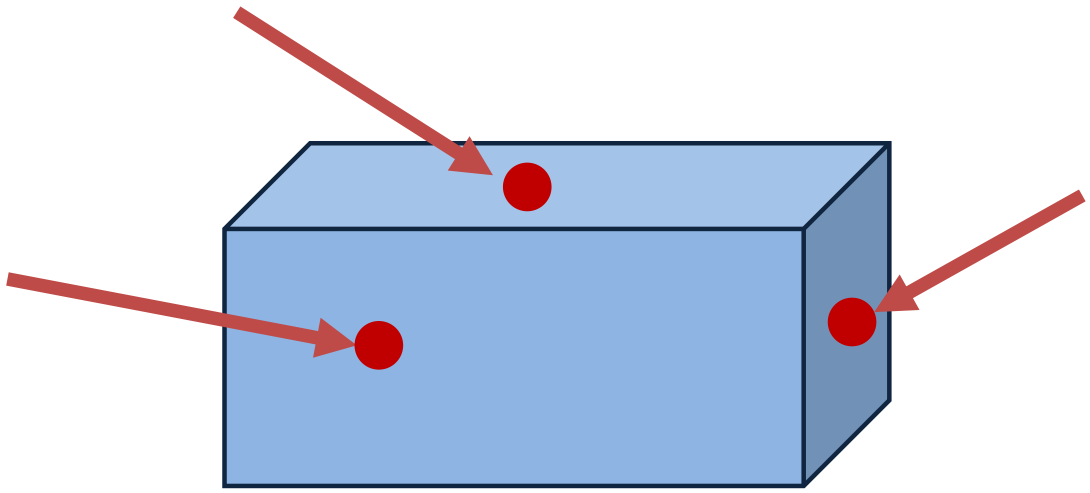
Grasp Planning

- Parameterizations:
  - Approach vector or wrist pose
  - Initial finger configuration
  - Contact-point specification

- Regardless of the parameterization, the **contact points define the quality of a grasp**

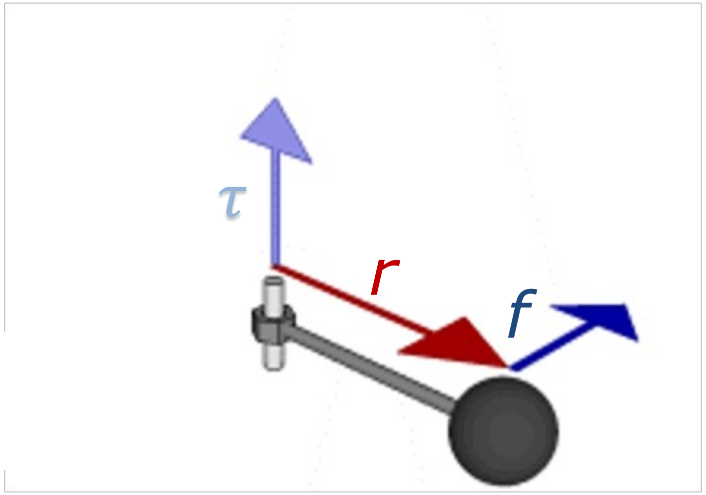
Grasp Modeling

- **Point-on-plane contact models:**
  Commonly used for grasping since the possible contact points for most objects are almost always on surfaces



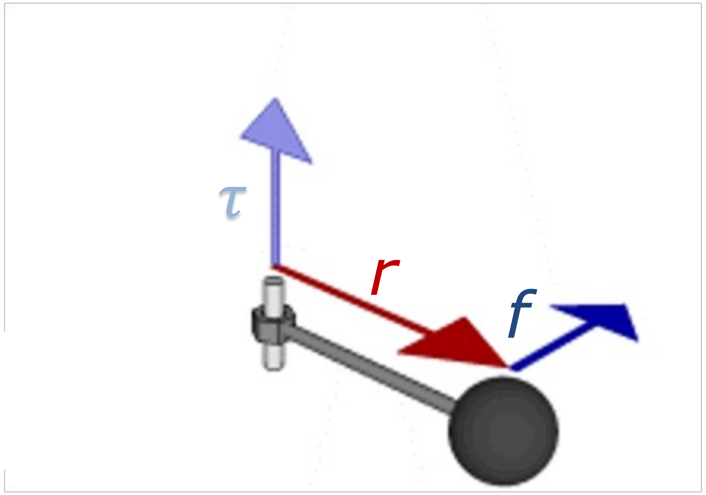
Basics: Force and Torque

- 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: $\tau = r \times f$



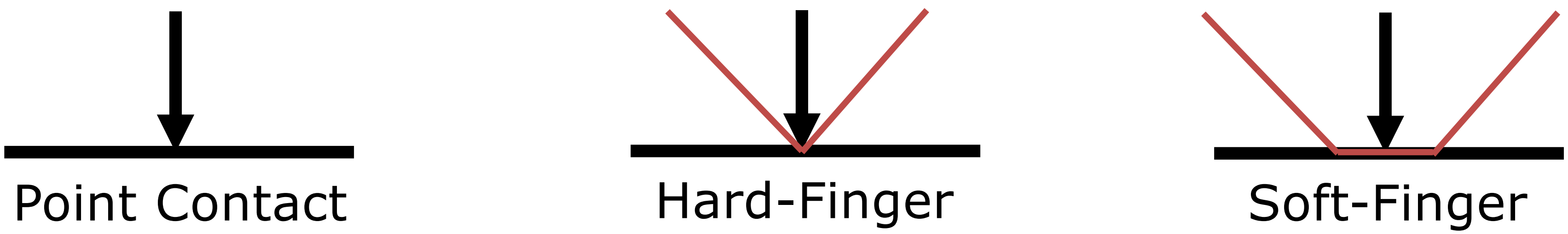
Basics: Force and Torque = Wrench

- Wrench $w = (f, \tau)$
- 6D vector



Point-on-Plane Contact Models

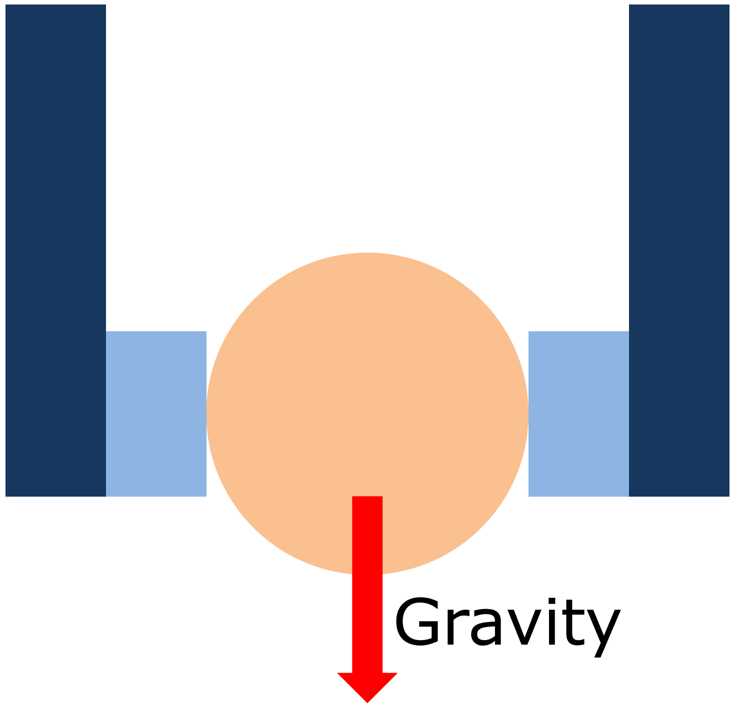
- **Frictionless point contact**:
  Forces can only be applied along the surface normal
- **Point contact with friction (hard-finger)**:
  Forces can be applied in directions other than just the surface normal, defined by a **friction cone**
- **Soft-finger contact**:
  Allows for torque around the surface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



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

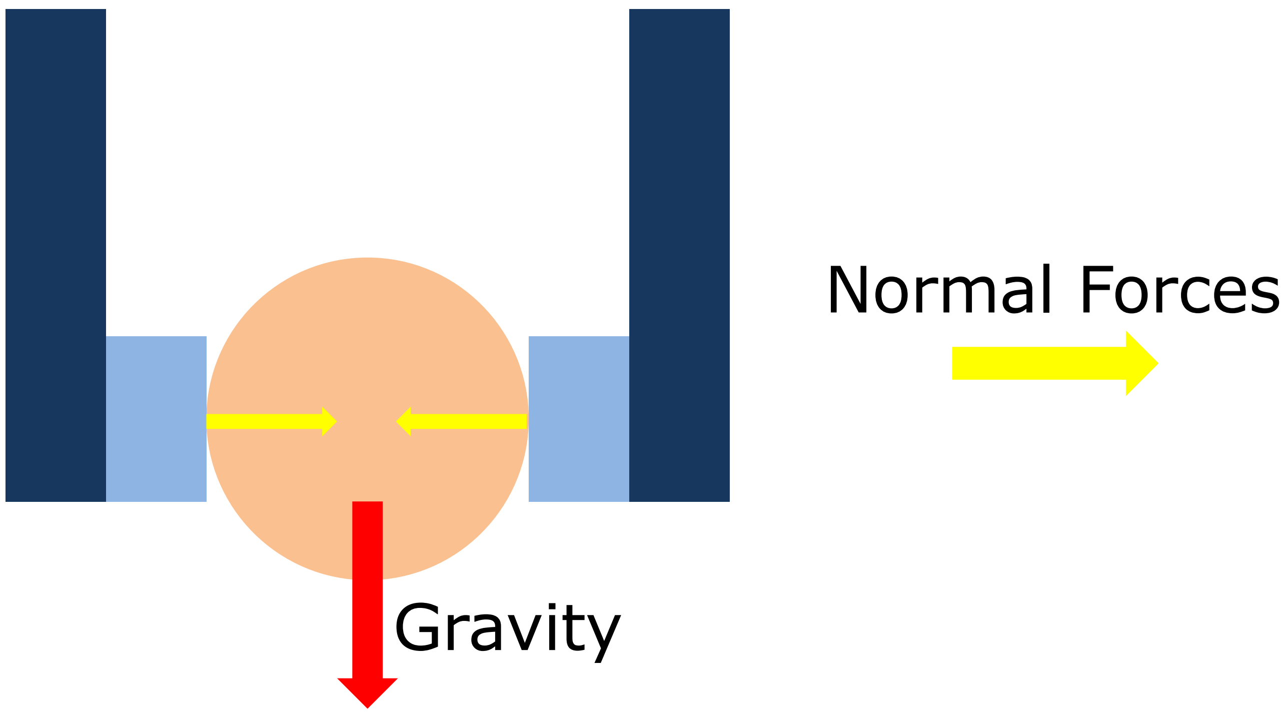
- How to **counter** gravity force to **lift an object**?

Robotiq



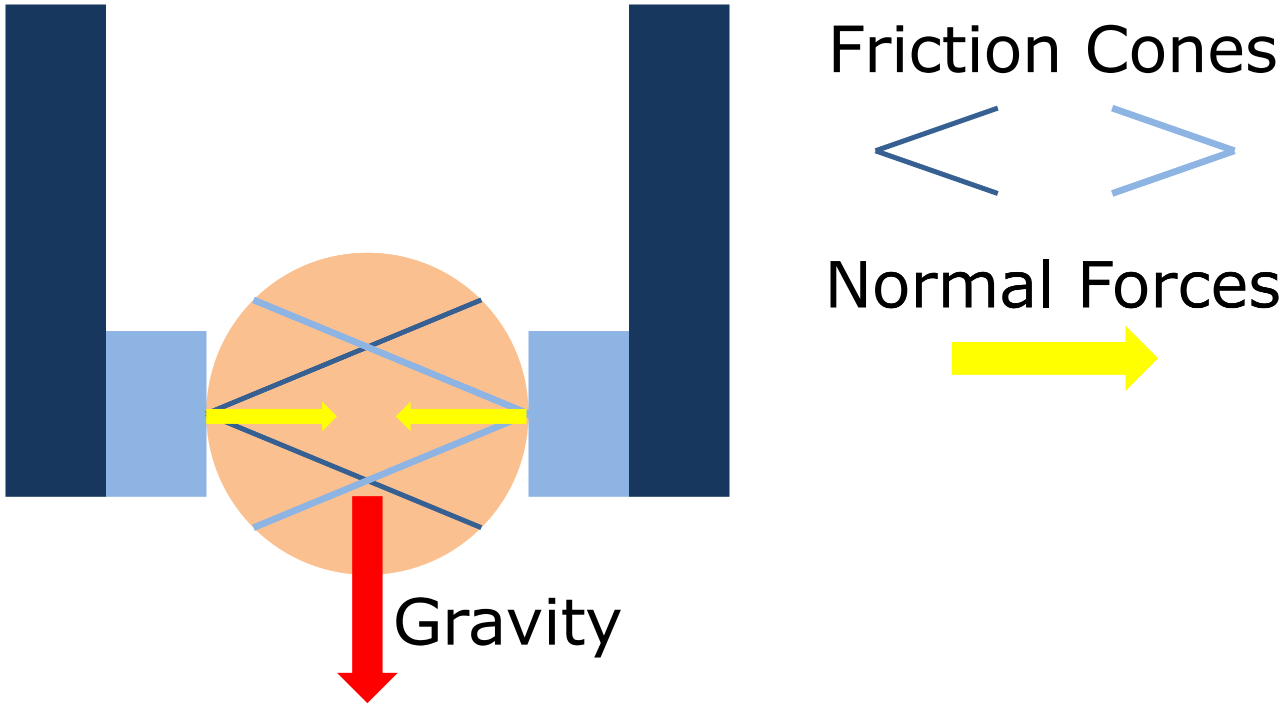
Parallel-Jaw (2-Finger) Grasping in 2D

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



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

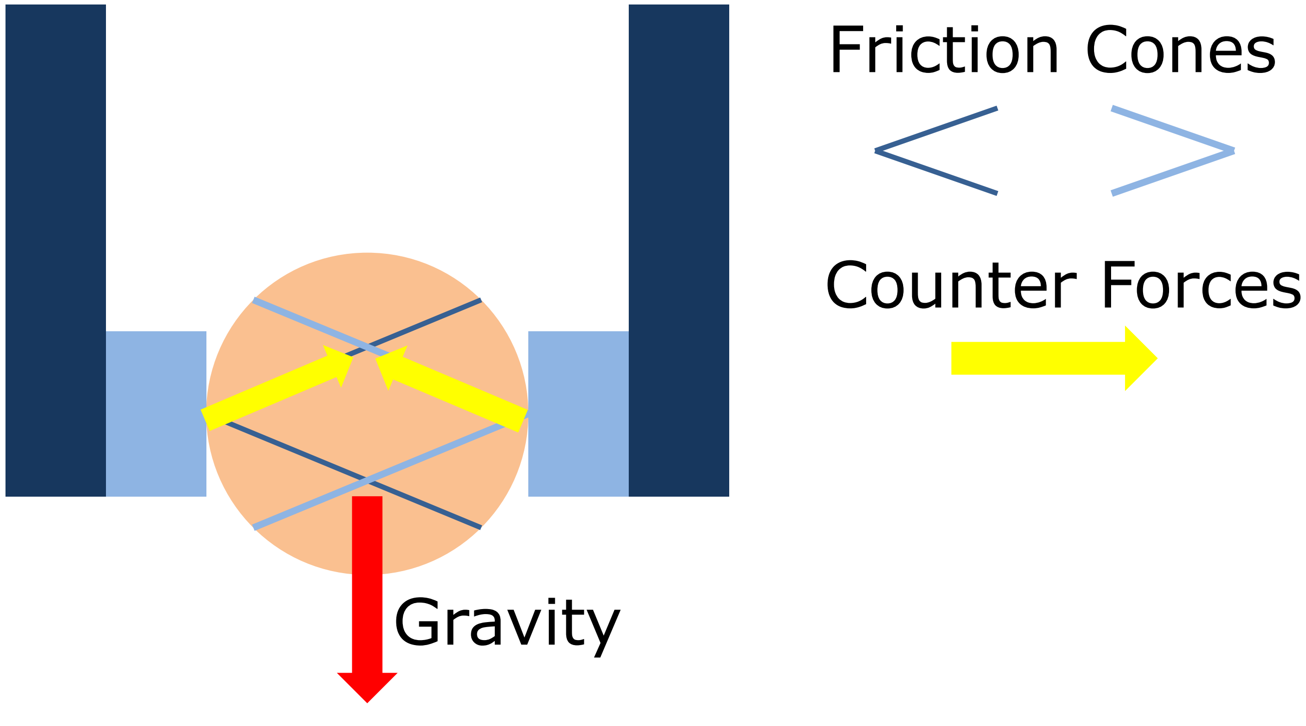
- How to **counter** gravity force to **lift an object**?



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

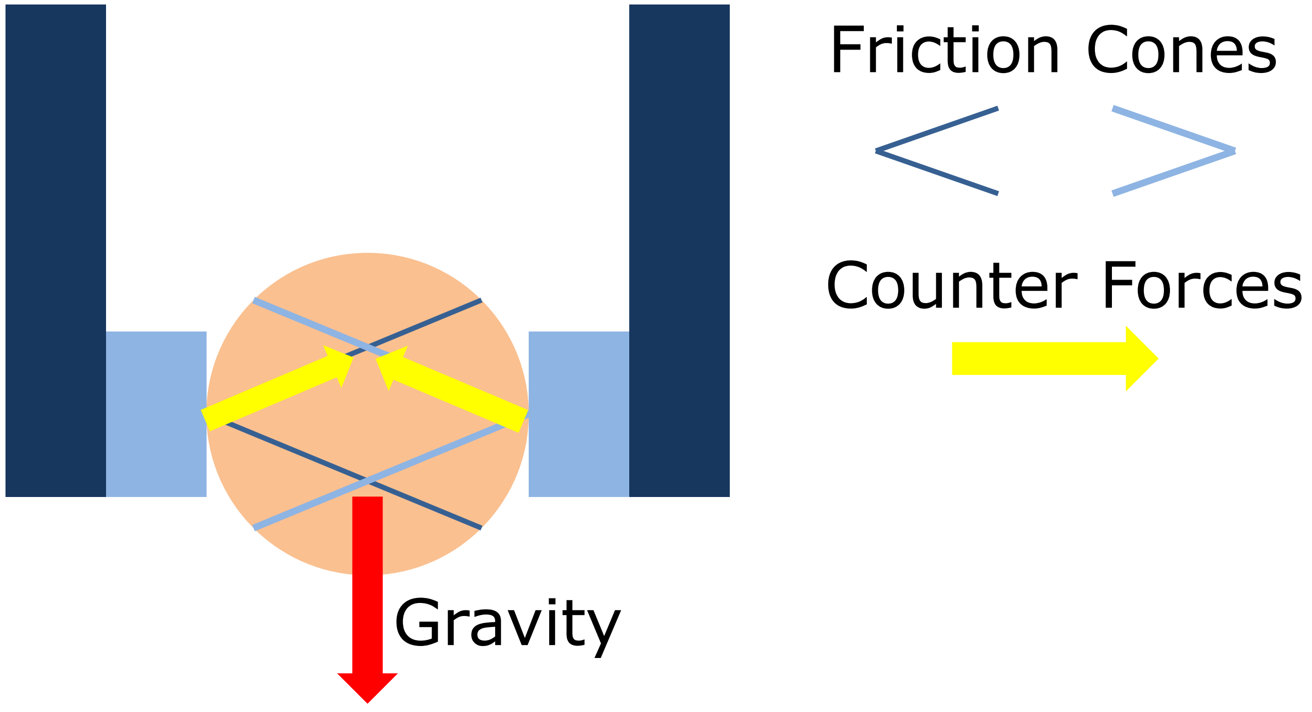
Force has to be greater than or equal to the gravitational force acting on the object



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

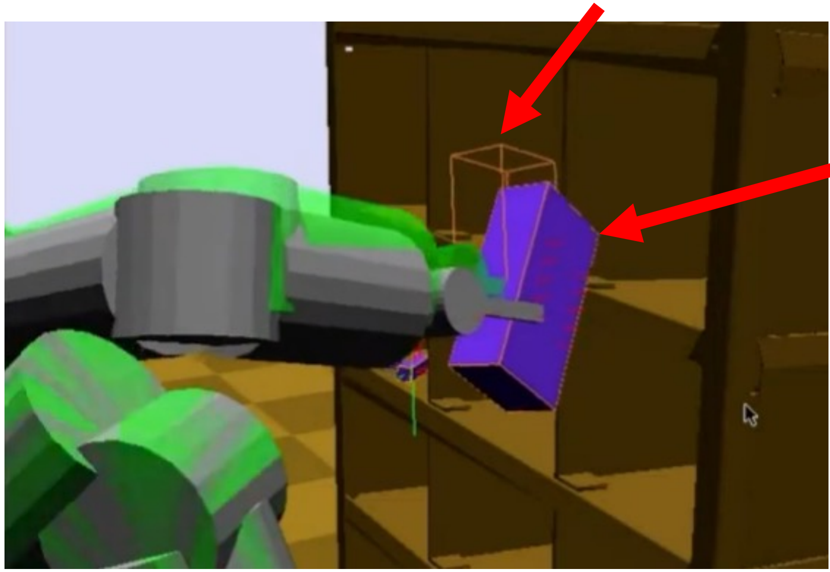
- How to **counter** gravity force to **lift an object**?
- In the friction cone, there are forces that counteract a given gravitational force

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



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

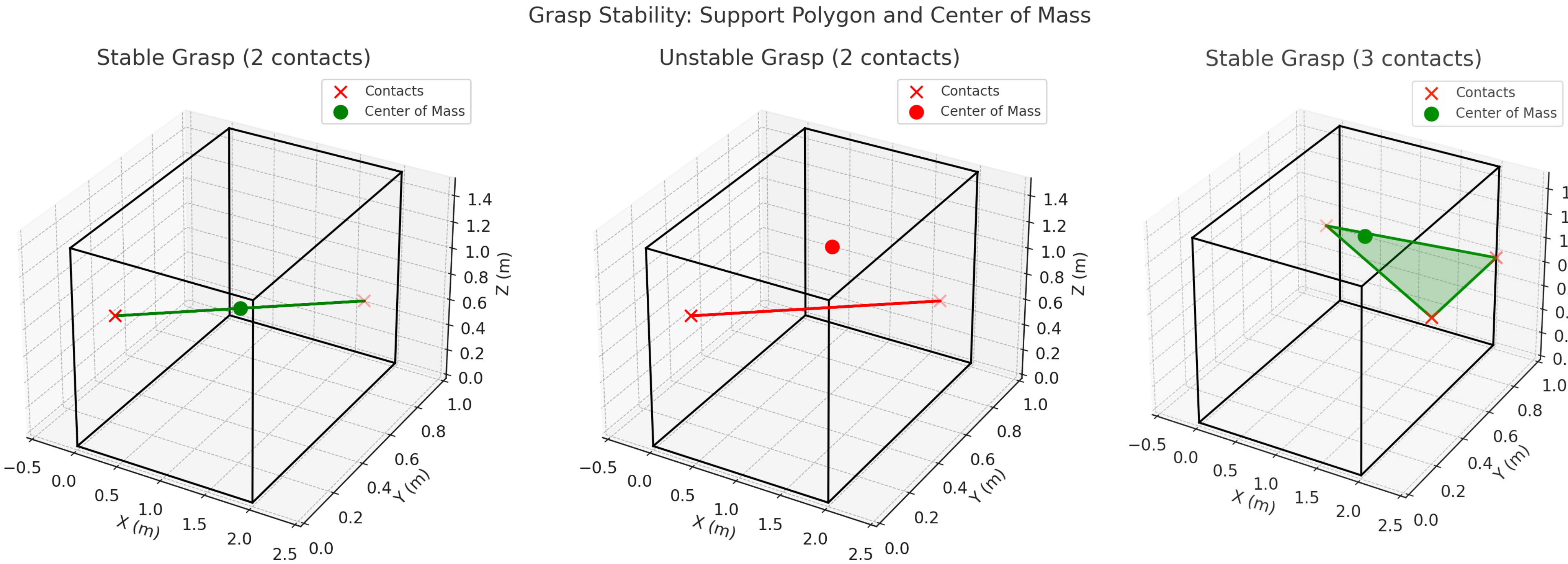
- With only two contact points, **unable to resist** torque about the axis
- This **may be also true** for more contact points



Courtesy: K. Hauser

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

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

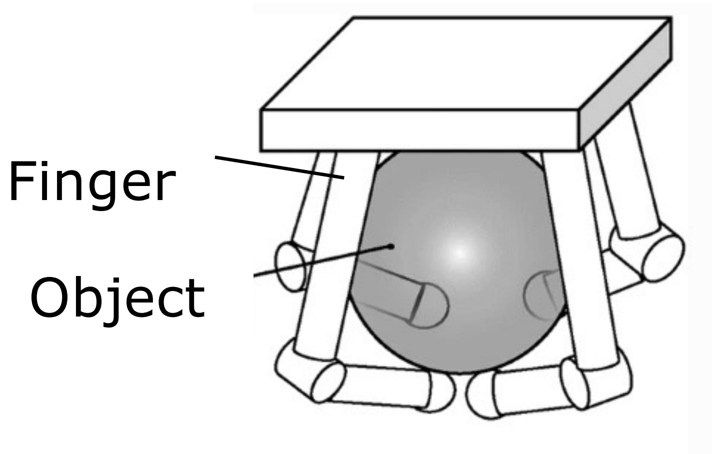


Grasp Quality Evaluation

- **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

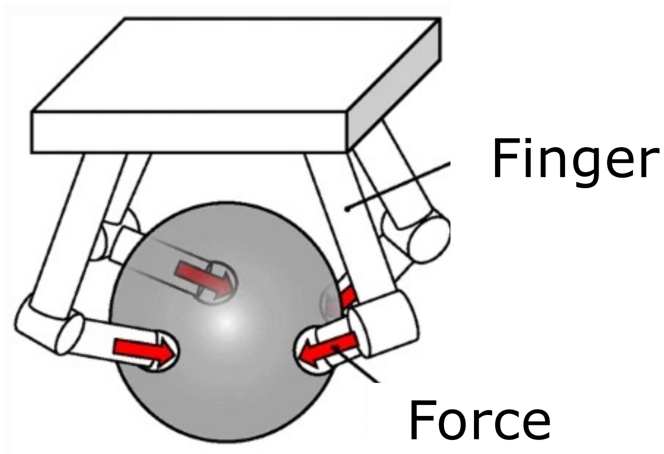
- Fixed grasp that **locks** the object between the fingers
- Geometric constraints alone prevent the object from moving
- No wrench (force + torque) can move the object
- Guaranteed immobilization
- But extremely sensitive to shape, requires many contacts
- Often not feasible to compute



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

Sampling-Based Grasp Planning

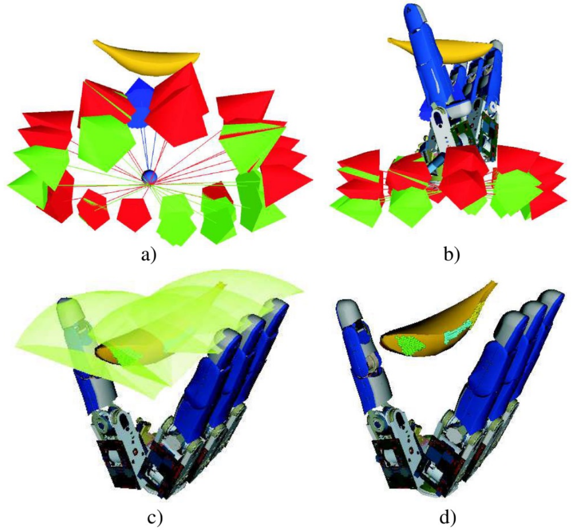
- Two approaches, both apply sampling and evaluate possible grasps
- Sample **candidate contacts** directly on object surface
- Use **predefined hand models** that come into contact when fingers close
  - Sample the location where the hand base will be placed
  - Simulate where the contacts happen after closing fingers

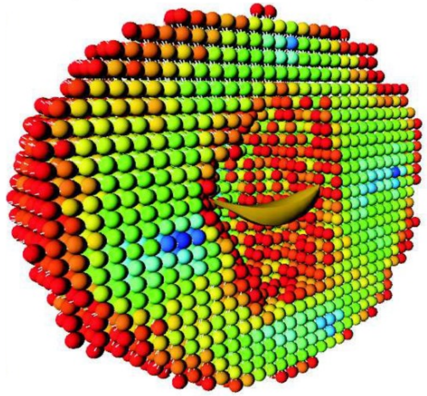
Sampling-Based Grasp Planning

- **Optimal grasps**: Combination of optimal contact points **and** optimal hand configuration
- Selection of **optimal contact points** on the object surface ignoring the actual hand geometry, can lead to contact locations unreachable for the real hand
- **Feasible hand configuration** can generate a weak grasp in the presence of small perturbations
- Grasp planning approaches try to satisfy both metrics

Offline Grasp Database Generation Using Known Hand and CAD Models

- **Graspability map:** offline generated grasps for known objects for given hand
- Multi-finger kinematics precomputed offline
- Store full hand-pose in database
- Requires **precise object model and pose**
- Fast database retrieval during execution
- Similar to reachability map concept (Ch.5)





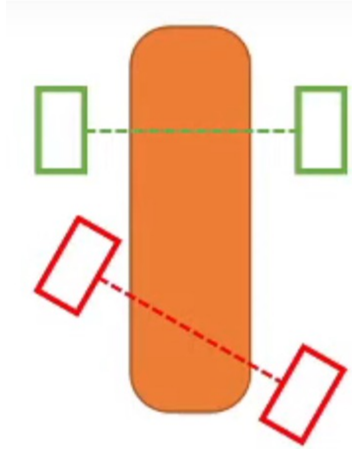
[Roa et al., IROS 2011]

Perception for Grasping

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

Heuristic Grasp Detection

- Sample antipodal grasps
  - Two opposing contact normals on object
  - Center c, axis a, width w
  - Sampling: **top-down** based on point cloud data
  - Scoring metrics based on normality, fit, clearance
- Sampling and scoring can become challenging for complex objects and without top-down restriction



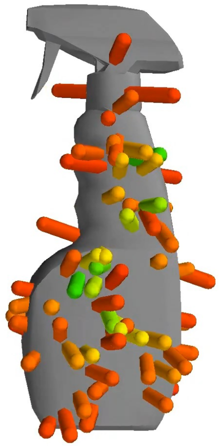
Hauser et al., Advanced Topics in Planning

Learning-Based Grasping Approaches

- **Sample & score:** synthetic grasps, analytic labels (DexNet, GPD)
- **Pixel regression:** real RGB-D, per-pixel labels (AnyGrasp)
- **Volumetric CNN:** TSDF volumes, voxel-wise labels (VGN)

Learning to Grasp

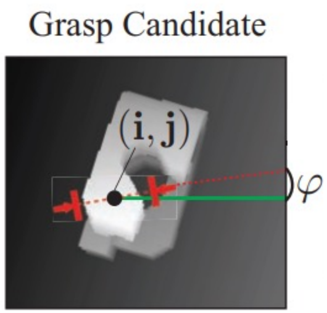
- To reduce **data collection time** of real objects for deep learning of robust robotic grasping, synthetic datasets can be used
- Example **Dex-Net 2.0**: Consists of 6.7 million point clouds, grasps, and analytic grasp metrics generated from thousands of 3D models



Mahler et al., Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, RSS 2017

Learning to Grasp

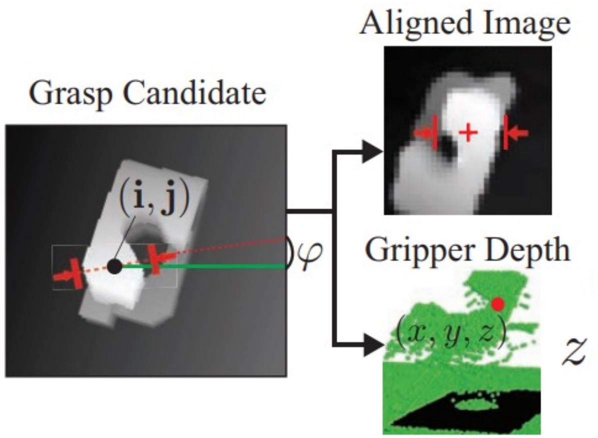
- Grasp Quality Convolutional Neural Network (GQ-CNN)
- Grasp candidates $u = (i, j, \phi, z)$ are generated from a **depth image** and transformed to align the image with the grasp center pixel (i, j) and orientation ϕ



Mahler et al., Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, RSS 2017

Learning to Grasp

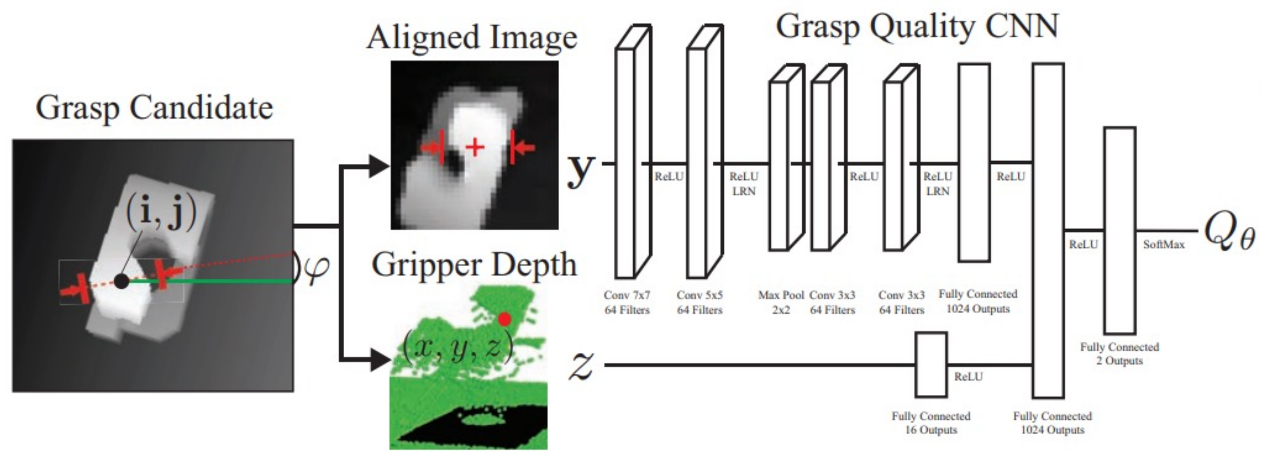
- Grasp Quality Convolutional Neural Network (GQ-CNN)
- Grasp candidates $u = (i, j, \phi, z)$ are generated from a **depth image** and transformed to align the image with the grasp center pixel (i, j) and orientation ϕ



Mahler et al., Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, RSS 2017

Learning to Grasp

- Depth edges as features for learning
- Network estimates the **probability of grasp success** $Q_\theta \in [0, 1]$ to rank grasp candidates
- Note: only top-down grasps



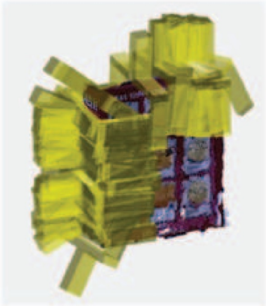
Mahler et al., Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, RSS 2017

Example

Mahler et al., Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics, RSS 2017

Grasp Pose Detection (GPD)

- **Input:** Raw point clouds
- Pipeline:
  - **Generate** candidate two-finger grasps by sampling surface point pairs
  - **Encode** local geometry into heightmap descriptors
  - **Score** each candidate 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

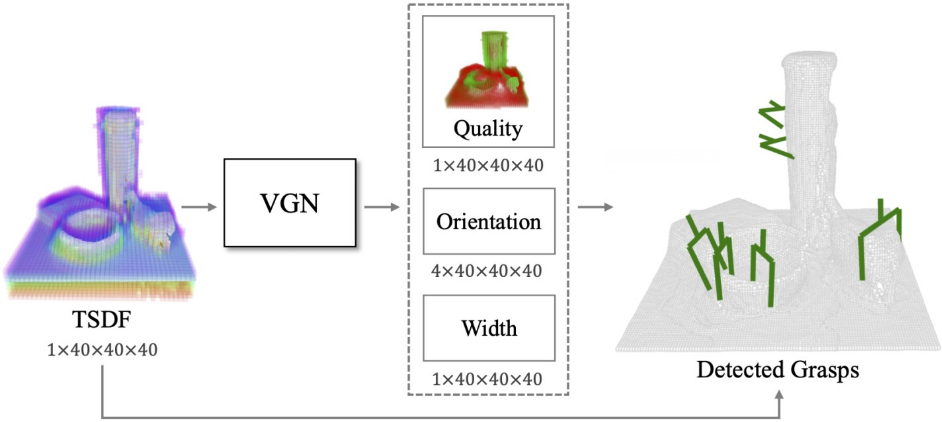
AnyGrasp

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

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

Volumetric Grasping Network (VGN)

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



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

GPD vs. AnyGrasp vs. VGN

Aspect	**GPD**	**AnyGrasp**	**VGN**
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
Multi-Finger Support	parallel-jaw only	parallel-jaw only	parallel-jaw only
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 • Modular	• Full 7-DoF • Dynamic tracking	• Ultra-fast • Leverages full 3D
Cons	• Slow • Static only	• Complex • Heavier than VGN and GPD	• Needs TSDF fusion • Black-box

Real-Time Grasp Correction with Perception

- Conventional approach: Plan once and go blind (open loop)
- Robust approach: Continuously update grasp pose during approach to correct for errors and disturbances
- **Sense:** Stream point cloud or RGB-D at each control cycle, fuse into TSDF if necessary (VGN)
- **Re-compute grasp:** Quickly 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
- Must coordinate finger trajectories and contact sequence

[Winkelbauer et al., IROS 2022]

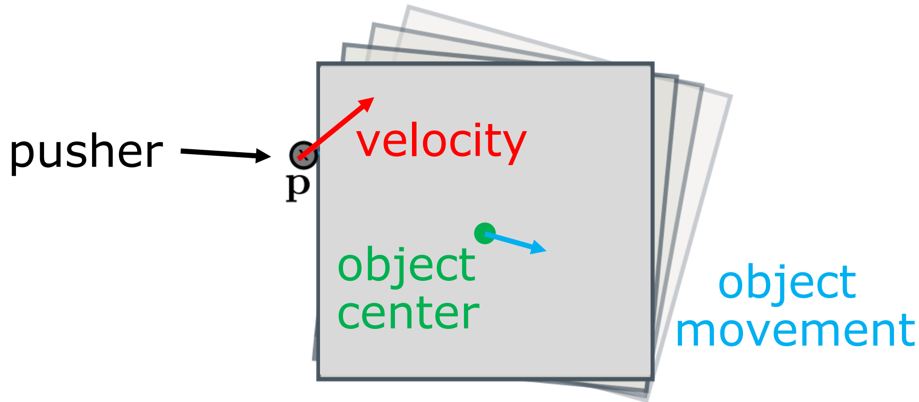
Pushing

Planar Pushing

- **Goal:** Control the pose of an object in 2D using only "pushing" contacts
- **Interaction** between the **object and surface** must also be considered
- Similar to physics-based contact models for grasping, such models can also be developed to **predict the sliding interactions** between objects and surface

Analytical Planar Pushing

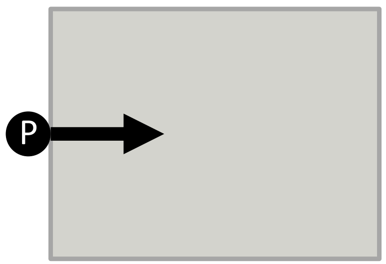
- Pushing modeled using an **analytical model**
- **Only approximate** and far from perfectly modeling the process of planar pushing
- **Predicts the object movement** given the pusher velocity, the contact point as well as mass, force, and friction-related parameters



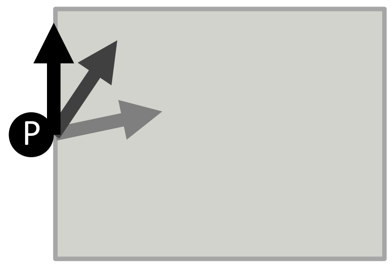
[Kloss et al., IJRR 2020]

Analytical Planar Pushing

- Predicting 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



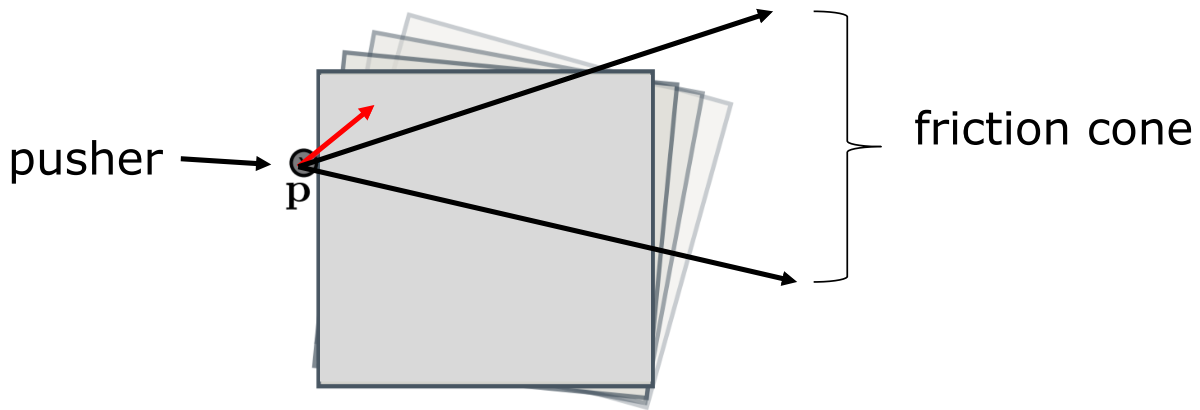
Sticking



Sliding

Analytical Planar Pushing

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



[Kloss et al., IJRR 2020]

Analytical Planar Pushing

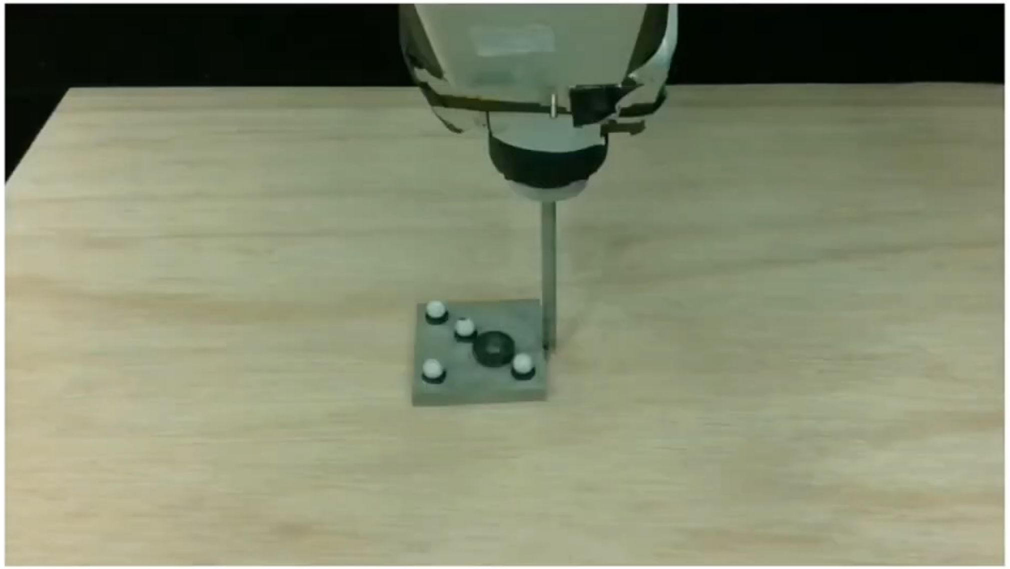
- **Motion cone**: Set of all object motions that keep the pushing contact **constant**
- If the push velocity is outside of the motion cone, the **contact will slide** but the object is still moved to some extend
- Otherwise, the contact is sticking and the **pusher velocity** is the effective **object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

Analytical Planar Pushing: Discussion

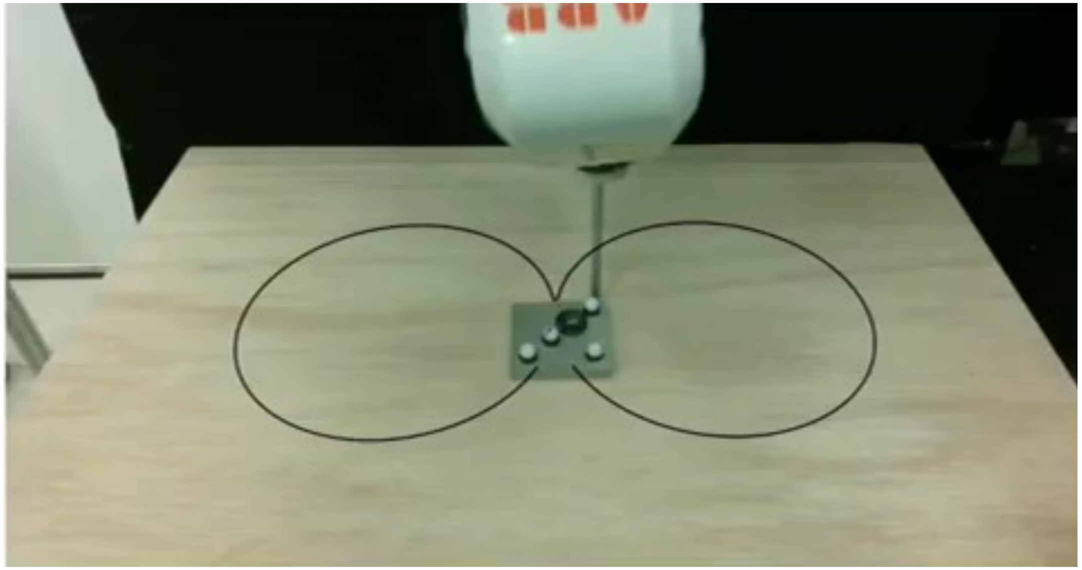
- Analytical model uses three **simplifying assumptions**:
  1. The force applied to the object is **big enough** to move the object, but **not to accelerate it**
  2. The pressure distribution of the object on the surface is **uniform**
  3. The friction between surface and object is **constant**
-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

Example

Data collection



Execution



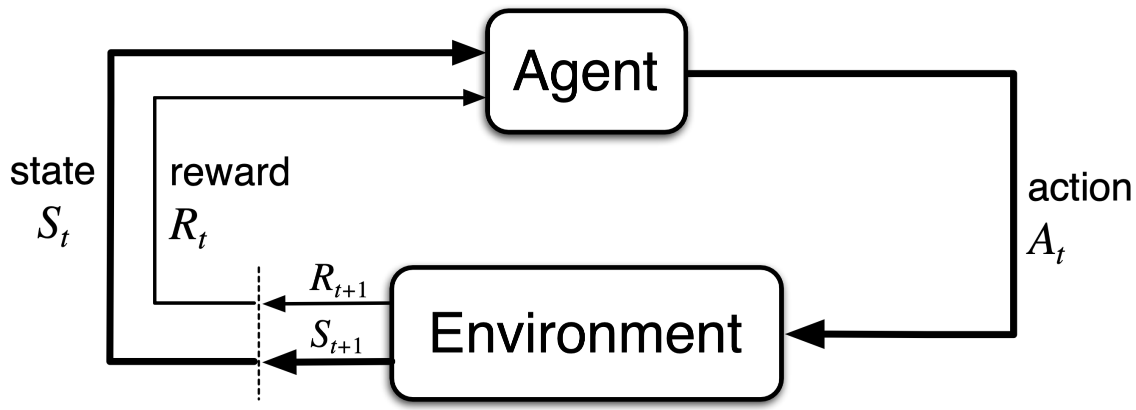
[Bauza et al., CORL 2019]

Learning to Push

- As for grasping, novel approaches do not rely on only analytical approaches 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

Reinforcement Learning (RL): Overview

- Reward hypothesis: Any goal can be formalized as the outcome of **maximizing a cumulative reward**
- **Goal**: pick the best action for any given state
- **Deep RL** introduces deep neural networks to solve RL problems

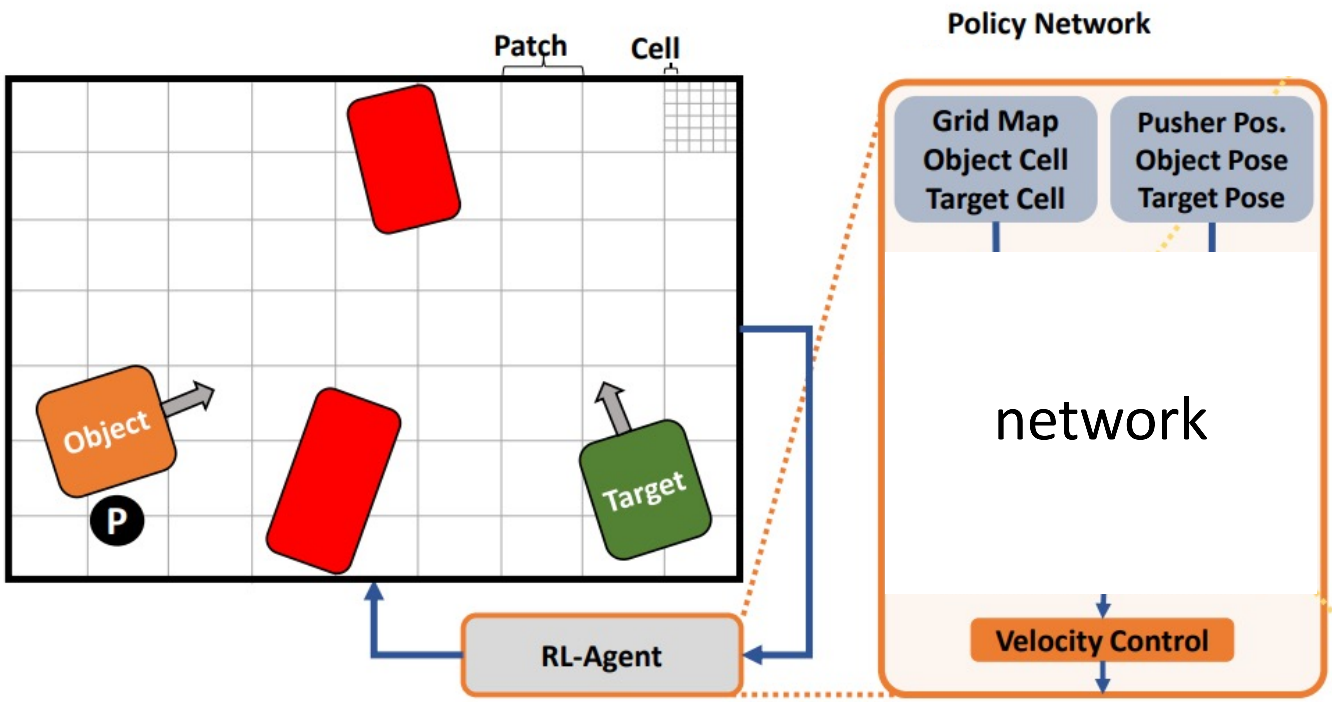


RL for Push Generation in Cluttered Scenes

- **Considered problem:**
  - Goal-oriented 2D pushing in cluttered environments
  - Avoiding contact with other objects

- **Approach:**
  - **Reinforcement learning** for obstacle avoiding pushing
  - **Attention mechanism** to select and focus on relevant features

RL for Push Generation in Cluttered Scenes

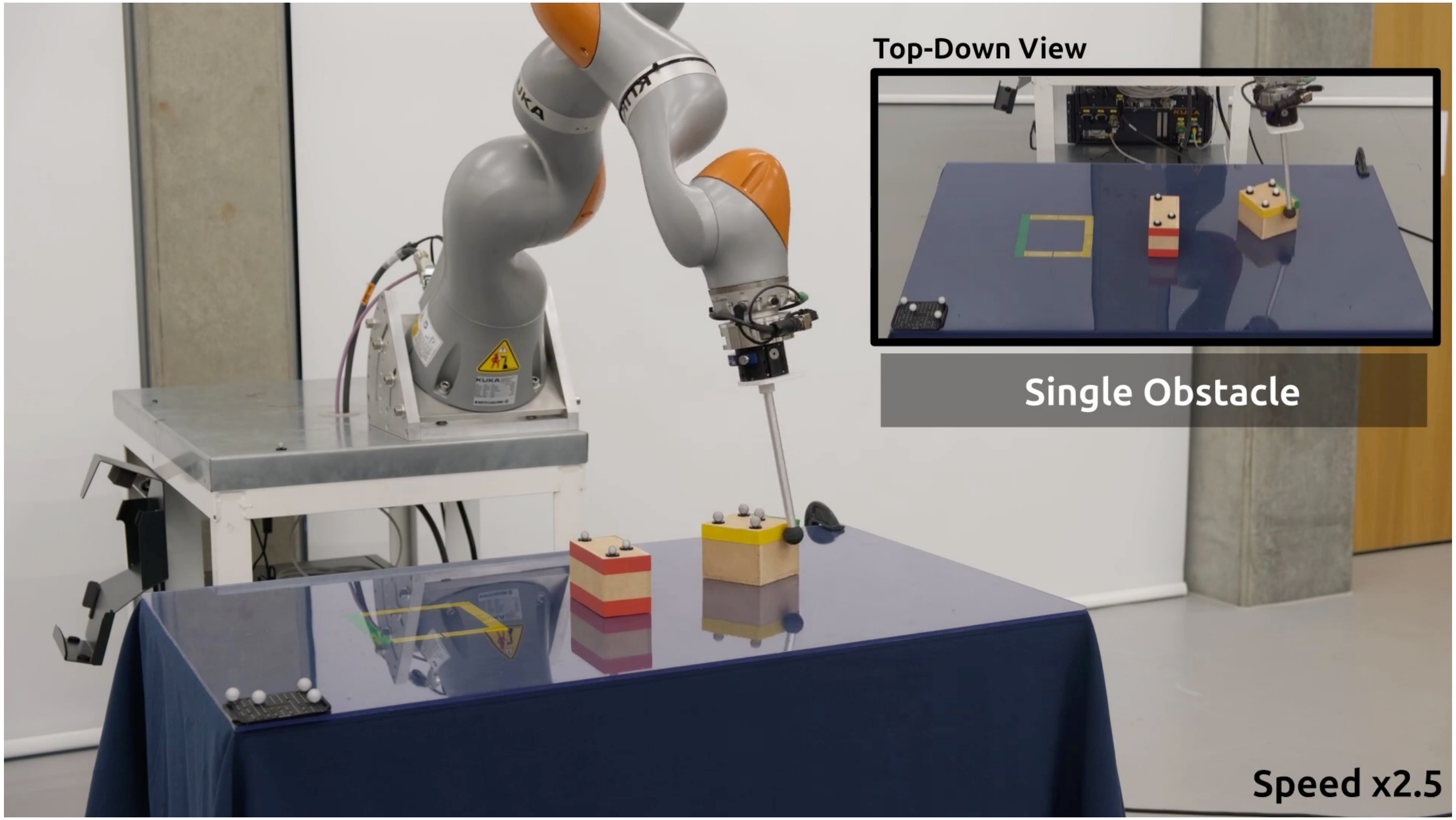


Dengler et al., Learning Goal-Directed Object Pushing in Cluttered Scenes with Location-Based Attention, IROS 2025

RL for Push Generation 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

Learned Object Pushing

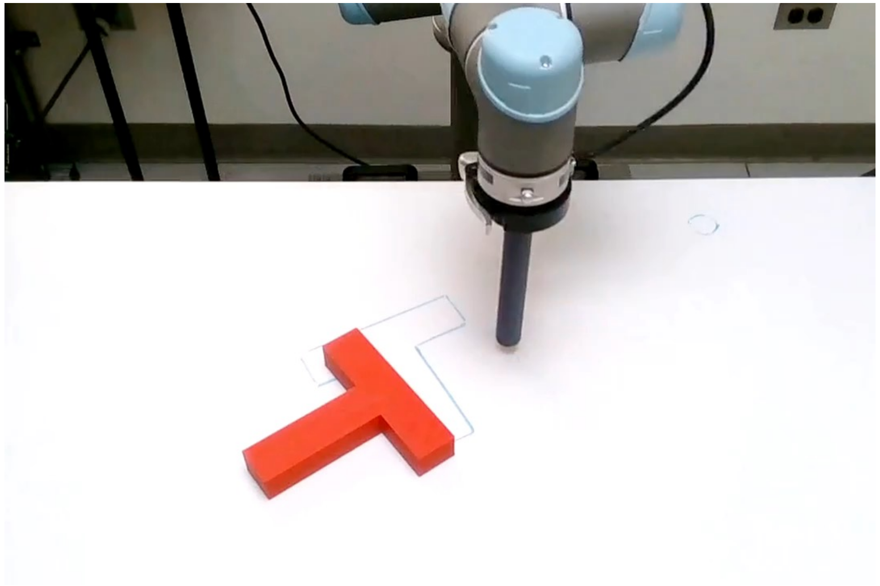


Dengler et al., Learning Goal-Directed Object Pushing in Cluttered Scenes with Location-Based Attention, IROS 2025

New Research Direction: Diffusion Policies

- 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/



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** analyses, **sampling-based planning**, large synthetic **datasets with analytic metrics**, and **deep learning** models 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:
  - **Reinforcement learning** using occupancy grids
  - **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
- *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