Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Institute for AI and Autonomous Systems

Cognition-Enabled Manipulation

An Overview

Dr. Alex Mitrevski
Master of Autonomous Systems

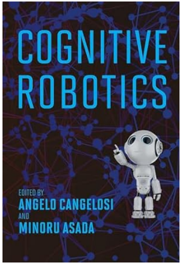
Structure


By Mustafa Ersen, Erhan Oztop, and Sanem Sariel

Cognition-Enabled Robot Manipulation in Human Environments

Service robots are expected to play an important role in our daily lives as our companions in home and work environments in the near future. An important requirement for fulfilling this expectation is to equip robots with skills to perform everyday manipulation tasks, the success of which is crucial for most home chores, such as cooking, cleaning, and shopping. Robots have been used successfully for manipulation tasks in well-structured and controlled factory environments for decades. Designing skills for robots working in uncontrolled human environments raises many potential challenges in various subdisciplines, such as computer vision, automated planning, and human–robot interaction. In spite of the recent progress in these fields, there are still challenges to tackle. This article outlines problems in different research areas related to mobile manipulation from the cognitive perspective, reviews recently published works and the state-of-the-art approaches to address these problems, and discusses open problems to be solved to realize robot assistants that can be used in manipulation tasks in unstructured human environments.

Requirements, Recent Work, and Open Problems

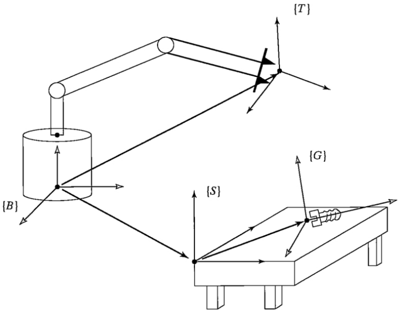


- Basics of robot manipulation
- Perception for cognition-enabled manipulation
- Learning-based robot manipulation
- Knowledge for manipulation
- Execution monitoring and failure recovery

Basics of Robot Manipulation

Object Pose Detection (Frames Revisited)

- Traditionally, manipulation involves a problem of finding:
  - **the pose of an object** that should be manipulated
  - **a transformation** that would bring the robot's end effector to the object
- But how should the robot actually move to that pose?



J. Craig, "Spatial Descriptions and Transformations," in *Introduction to Robotics: Mechanics and Control*. Pearson Education, Inc. 2005, ch. 2, p. 39.

Motion Planning

Traditional research in manipulation has been concerned with the following aspects:

- **Path planning**: Finding a path — a sequence of poses — that brings the end effector from its current pose to the goal pose
- **Trajectory planning**: Finding a fully specified trajectory — a sequence of poses and velocities — that brings the end effector to the goal pose
- **Low-level control**: Deciding how to actually move the robot's actuators

Given:

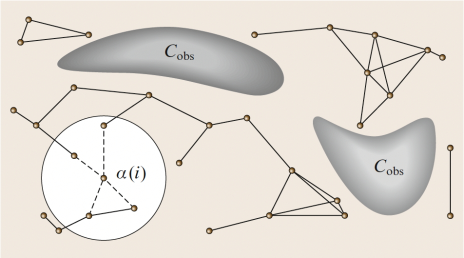
1. A *workspace* $\mathcal{W}$, where either $\mathcal{W}=\mathbb{R}^2$ or $\mathcal{W}=\mathbb{R}^3$.
2. An *obstacle region* $\mathcal{O}\subset\mathcal{W}$.
3. A *robot* defined in $\mathcal{W}$. Either a rigid body $\mathcal{A}$ or a collection of m links: $\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_m$.
4. The *configuration space* $\mathcal{C}$ ($\mathcal{C}_{\text{obs}}$ and $\mathcal{C}_{\text{free}}$ are then defined).
5. An *initial configuration* $\boldsymbol{q}_{\text{I}}\in\mathcal{C}_{\text{free}}$.
6. A *goal configuration* $\boldsymbol{q}_{\text{G}}\in\mathcal{C}_{\text{free}}$. The initial and goal configuration are often called a *query* $(\boldsymbol{q}_{\text{I}},\boldsymbol{q}_{\text{G}})$.

Compute a (continuous) path, $\tau:[0,1]\to\mathcal{C}_{\text{free}}$, *such that* $\tau(0)=\boldsymbol{q}_{\text{I}}$ *and* $\tau(1)=\boldsymbol{q}_{\text{G}}$.

L. E. Kavraki and S. M. LaValle, "Motion Planning," in *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg. 2008, ch. 5, p. 111. Available: https://link.springer.com/book/10.1007/978-3-540-30301-5

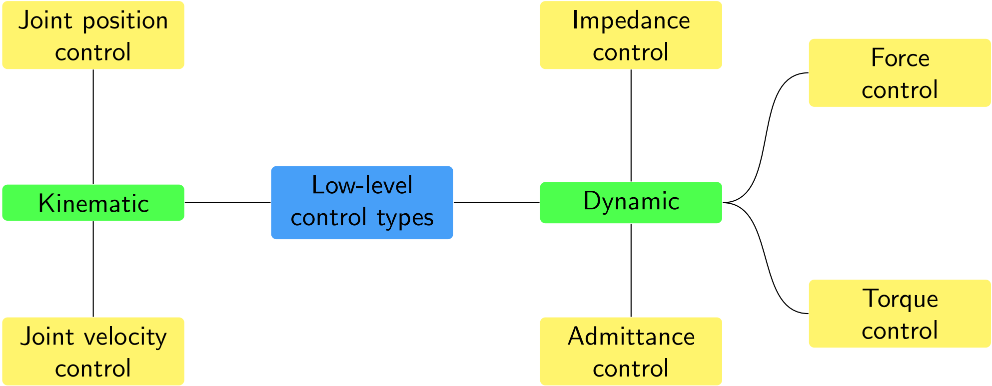
Roadmap Planning

- One commonly used procedure of finding paths from the start to the goal configuration is that of **(probabilistic) roadmap planning**
- This procedure looks for valid paths in configuration space by **exploring local neighbourhoods**, thereby expanding the path further **until the start and goal configuration are connected**



L. E. Kavraki and S. M. LaValle, "Motion Planning," in *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg. 2008, ch. 5, p. 111. Available: https://link.springer.com/book/10.1007/978-3-540-30301-5

Low-Level Robot Control



Is Traditional Manipulation Cognition-Enabled?

Not really, as it does not take into account:

Is Traditional Manipulation Cognition-Enabled?

Not really, as it does not take into account:

Knowledge about objects (e.g. physical properties) and adaptivity based on that

Is Traditional Manipulation Cognition-Enabled?

Not really, as it does not take into account:

Knowledge about objects (e.g. physical properties) and adaptivity based on that

Prior manipulation experiences — every problem instance is treated as a separate manipulation episode

Is Traditional Manipulation Cognition-Enabled?

Not really, as it does not take into account:

Knowledge about objects (e.g. physical properties) and adaptivity based on that

Prior manipulation experiences — every problem instance is treated as a separate manipulation episode

The existence of other agents in the world — generally assumes that the environment is static and that only the robot can change the state of an object (although there are dynamic planners as well)

Is Traditional Manipulation Cognition-Enabled?

Not really, as it does not take into account:

Knowledge about objects (e.g. physical properties) and adaptivity based on that

Prior manipulation experiences — every problem instance is treated as a separate manipulation episode

The existence of other agents in the world — generally assumes that the environment is static and that only the robot can change the state of an object (although there are dynamic planners as well)

Information about human-acceptable trajectories — in principle, a motion planner accepts any valid solution (acceptability objectives can be optimised during planning, but may be difficult to specify)

What is Cognition-Enabled Manipulation Then?

- We can talk about cognition-enabled manipulation **if a robot is able to consume different sources of knowledge about the environment** that it is interacting with

What is Cognition-Enabled Manipulation Then?

- We can talk about cognition-enabled manipulation **if a robot is able to consume different sources of knowledge about the environment** that it is interacting with
- **The ability to improve its skills based on experience** as well as to **learn new skills** is also an important aspect of cognition-enabled manipulation

What is Cognition-Enabled Manipulation Then?

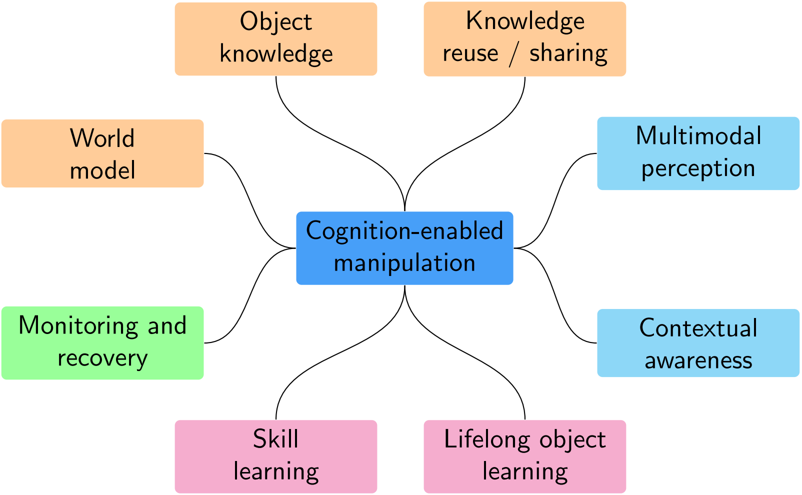
- We can talk about cognition-enabled manipulation **if a robot is able to consume different sources of knowledge about the environment** that it is interacting with
- **The ability to improve its skills based on experience** as well as to **learn new skills** is also an important aspect of cognition-enabled manipulation
- Particularly in human-centred environments, a cognitive robot should also be able to **move in a predictable / "human-like" way**

What is Cognition-Enabled Manipulation Then?

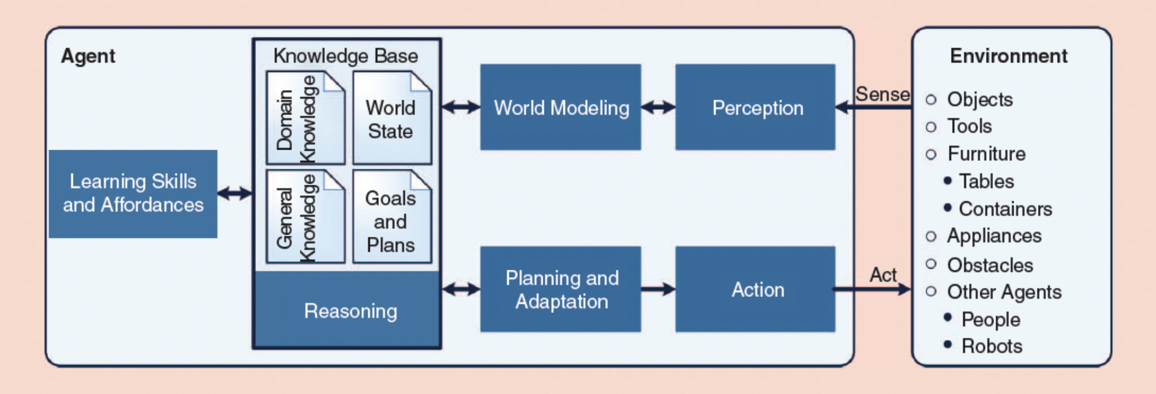
- We can talk about cognition-enabled manipulation **if a robot is able to consume different sources of knowledge about the environment** that it is interacting with
- **The ability to improve its skills based on experience** as well as to **learn new skills** is also an important aspect of cognition-enabled manipulation
- Particularly in human-centred environments, a cognitive robot should also be able to **move in a predictable / "human-like" way**

Cognition-enabled manipulation enables a robot to interact with the environment by taking into account all available environment information, to move in a human-like way, to monitor its execution and recover from failures appropriately, as well as to acquire and improve its skills through learning

Elements of Cognition-Enabled Manipulation



Elements of Cognition-Enabled Manipulation



Perception for Cognition-Enabled Manipulation

Perception Capabilities for Cognitive Robots

Cognition-enabled manipulation can be facilitated by a perceptual system that enables capabilities as:

Grounding symbols to real-world entities

Using information about the task context to inform both perceptual and manipulation activities

Recognising and using semantic environment information (e.g. room recognition)

Incorporating multimodal perceptual information (such as visual, auditory, and tactile information)

Recognising new / unknown objects (no closed-world assumption)

Symbol Grounding

- In cognitivist and hybrid systems, **symbols that are used to represent knowledge about the world in the robot's knowledge base need to be mapped to real entities in the world**

[1] S. Coradeschi et al., "A Short Review of Symbol Grounding in Robotic and Intelligent Systems," *KI - Künstliche Intelligenz*, vol. 27, pp. 129-136, 2013. Available: https://doi.org/10.1007/s13218-013-0247-2

Symbol Grounding

- In cognitivist and hybrid systems, **symbols that are used to represent knowledge about the world in the robot's knowledge base need to be mapped to real entities in the world**

- The symbol grounding problem[1] is concerned with **how the mapping of symbols can be done** (note that a ground expression is one without free variables)

[1] S. Coradeschi et al., "A Short Review of Symbol Grounding in Robotic and Intelligent Systems," *KI - Künstliche Intelligenz*, vol. 27, pp. 129-136, 2013. Available: https://doi.org/10.1007/s13218-013-0247-2

Symbol Grounding

- In cognitivist and hybrid systems, **symbols that are used to represent knowledge about the world in the robot's knowledge base need to be mapped to real entities in the world**
- The symbol grounding problem[1] is concerned with **how the mapping of symbols can be done** (note that a ground expression is one without free variables)
- Consider the following expression, which might define an action for when some robot R can pick an object X that is currently on a surface T with its hand H:

$$free(R, H) \wedge inFrontOf(R, T) \wedge on(X, T) \implies pick(R, H, X)$$

[1] S. Coradeschi et al., "A Short Review of Symbol Grounding in Robotic and Intelligent Systems," *KI - Künstliche Intelligenz*, vol. 27, pp. 129-136, 2013. Available: https://doi.org/10.1007/s13218-013-0247-2

Symbol Grounding

- In cognitivist and hybrid systems, **symbols that are used to represent knowledge about the world in the robot's knowledge base need to be mapped to real entities in the world**
- The symbol grounding problem[1] is concerned with **how the mapping of symbols can be done** (note that a ground expression is one without free variables)
- Consider the following expression, which might define an action for when some robot R can pick an object X that is currently on a surface T with its hand H:

$$free(R,H) \wedge inFrontOf(R,T) \wedge on(X,T) \implies pick(R,H,X)$$

In a ground version of the expression, all variables would be assigned to real-world entities:

$$free(lucy,hand) \wedge inFrontOf(lucy,desk) \wedge on(cup,desk) \implies pick(lucy,hand,cup)$$

[1] S. Coradeschi et al., "A Short Review of Symbol Grounding in Robotic and Intelligent Systems," *KI - Künstliche Intelligenz*, vol. 27, pp. 129-136, 2013. Available: https://doi.org/10.1007/s13218-013-0247-2

Symbol Grounding

- In cognitivist and hybrid systems, **symbols that are used to represent knowledge about the world in the robot's knowledge base need to be mapped to real entities in the world**
- The symbol grounding problem[1] is concerned with **how the mapping of symbols can be done** (note that a ground expression is one without free variables)
- Consider the following expression, which might define an action for when some robot R can pick an object X that is currently on a surface T with its hand H:

$$free(R, H) \land inFrontOf(R, T) \land on(X, T) \implies pick(R, H, X)$$

  In a ground version of the expression, all variables would be assigned to real-world entities:

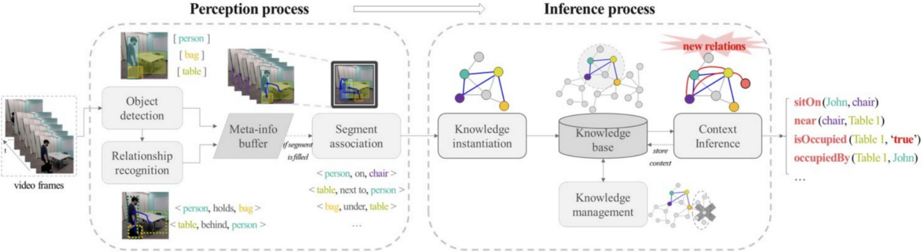
$$free(lucy, hand) \land inFrontOf(lucy, desk) \land on(cup, desk) \implies pick(lucy, hand, cup)$$

- Grounding is commonly done using connectionist models (particularly deep neural networks on more modern systems)

[1] S. Coradeschi et al., "A Short Review of Symbol Grounding in Robotic and Intelligent Systems," *KI - Künstliche Intelligenz*, vol. 27, pp. 129-136, 2013. Available: https://doi.org/10.1007/s13218-013-0247-2

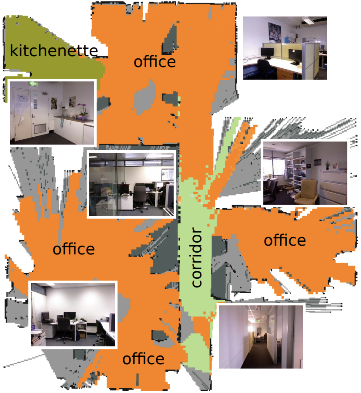
Contextual Awareness

- The ability to **recognise the execution context and act in a context-aware manner** is one important characteristic of cognitive manipulation (e.g. is the robot interacting with a sick patient or a healthy person?)
- Contextual information is an important prerequisite for personalising robot behaviour
- Difficult problem: **Which contextual information is relevant to attend to?**



D. Chang and B. Han, "Knowledge-based Visual Context-Aware Framework for Applications in Robotic Services," in *Proc. IEEE/CVF Winter Conf. Applications of Computer Vision*, 2023, pp. 70-78.

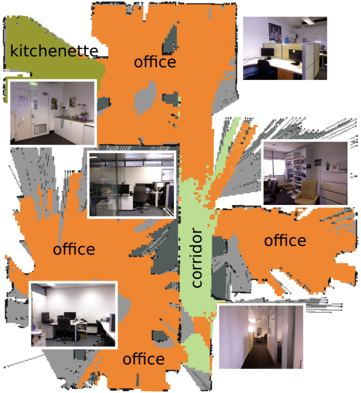
Semantic Mapping and Localisation



N. Sünderhauf et al., "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016, pp. 5729–5736. Available: https://doi.org/10.1109/ICRA.2016.7487796

- One way to bring context into the perceptual process is to use a **semantic map**
- A semantic map **assigns semantic meaning to places (or items) based on perceptual features**

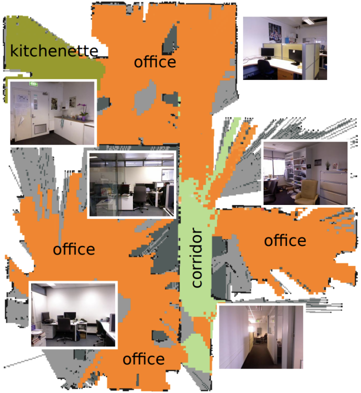
Semantic Mapping and Localisation



N. Sünderhauf et al., "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016, pp. 5729–5736. Available: https://doi.org/10.1109/ICRA.2016.7487796

- One way to bring context into the perceptual process is to use a **semantic map**
- A semantic map **assigns semantic meaning to places (or items) based on perceptual features**
- Semantic information about the robot's location can be used to:

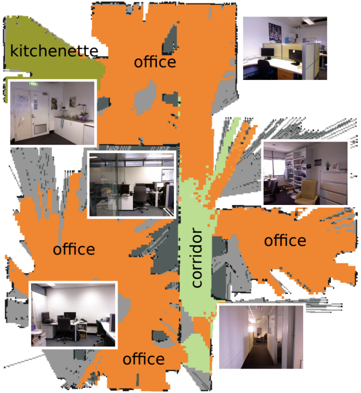
Semantic Mapping and Localisation



N. Sünderhauf et al., "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016, pp. 5729–5736. Available: https://doi.org/10.1109/ICRA.2016.7487796

- One way to bring context into the perceptual process is to use a **semantic map**
- A semantic map **assigns semantic meaning to places (or items) based on perceptual features**
- Semantic information about the robot's location can be used to:
  - **Inform subsequent perceptual actions** (e.g. if a robot is looking for an object)

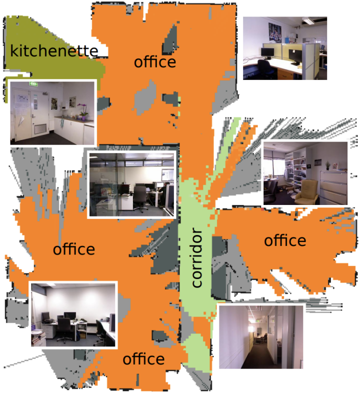
Semantic Mapping and Localisation



N. Sünderhauf et al., "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016, pp. 5729–5736. Available: https://doi.org/10.1109/ICRA.2016.7487796

- One way to bring context into the perceptual process is to use a **semantic map**
- A semantic map **assigns semantic meaning to places (or items) based on perceptual features**
- Semantic information about the robot's location can be used to:
  - **Inform subsequent perceptual actions** (e.g. if a robot is looking for an object)
  - **Guide the execution of manipulation skills** (e.g. to prevent unsafe motions close to patients in a room)

Semantic Mapping and Localisation



N. Sünderhauf et al., "Place categorization and semantic mapping on a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016, pp. 5729–5736. Available: https://doi.org/10.1109/ICRA.2016.7487796

- One way to bring context into the perceptual process is to use a **semantic map**
- A semantic map **assigns semantic meaning to places (or items) based on perceptual features**
- Semantic information about the robot's location can be used to:
  - **Inform subsequent perceptual actions** (e.g. if a robot is looking for an object)
  - **Guide the execution of manipulation skills** (e.g. to prevent unsafe motions close to patients in a room)
  - **Detect execution anomalies** (e.g. to recognise that a bottle of medicine is on a coffee table where children can reach it)

Multimodal Perception

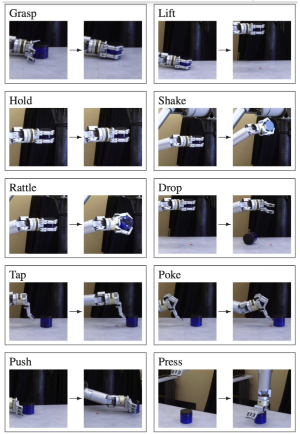


Fig. 2: Before and after images of the 10 exploratory behaviors that the robot used to learn about the objects.

J. Sinapov et al., "Learning relational object categories using behavioral exploration and multimodal perception," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, pp. 5691–5698. Available: https://doi.org/10.1109/ICRA.2014.6907696

- In traditional applications, a robot uses one main modality to perceive its environment (e.g. a lidar for localisation, a camera for object detection); **a multimodal perception system combines different modalities** (e.g. visual and tactile information)

Multimodal Perception

Fig. 2: Before and after images of the 10 exploratory behaviors that the robot used to learn about the objects.

J. Sinapov et al., "Learning relational object categories using behavioral exploration and multimodal perception," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, pp. 5691–5698. Available: https://doi.org/10.1109/ICRA.2014.6907696

- In traditional applications, a robot uses one main modality to perceive its environment (e.g. a lidar for localisation, a camera for object detection); **a multimodal perception system combines different modalities** (e.g. visual and tactile information)
- Multimodality is particularly important for **exploring the environment efficiently**, but also for **improving the execution robustness**

Multimodal Perception

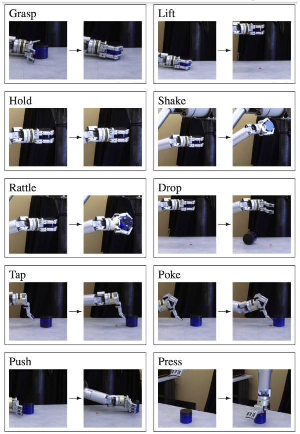


Fig. 2: Before and after images of the 10 exploratory behaviors that the robot used to learn about the objects.

J. Sinapov et al., "Learning relational object categories using behavioral exploration and multimodal perception," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, pp. 5691–5698. Available: https://doi.org/10.1109/ICRA.2014.6907696

- In traditional applications, a robot uses one main modality to perceive its environment (e.g. a lidar for localisation, a camera for object detection); **a multimodal perception system combines different modalities** (e.g. visual and tactile information)
- Multimodality is particularly important for **exploring the environment efficiently**, but also for **improving the execution robustness**
- Multimodality can be achieved using **manual heuristics** (classical approach) or with **multimodal neural networks** (more recent approach)

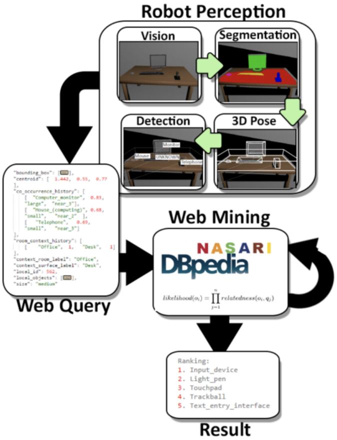
Multimodal Perception

Fig. 2: Before and after images of the 10 exploratory behaviors that the robot used to learn about the objects.

J. Sinapov et al., "Learning relational object categories using behavioral exploration and multimodal perception," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, pp. 5691–5698. Available: https://doi.org/10.1109/ICRA.2014.6907696

- In traditional applications, a robot uses one main modality to perceive its environment (e.g. a lidar for localisation, a camera for object detection); **a multimodal perception system combines different modalities** (e.g. visual and tactile information)
- Multimodality is particularly important for **exploring the environment efficiently**, but also for **improving the execution robustness**
- Multimodality can be achieved using **manual heuristics** (classical approach) or with **multimodal neural networks** (more recent approach)
- Multimodal perception is indispensable for human manipulation as well (self-experiment: put on gloves — that affect your tactile sensing — and try to grasp a bottle with your eyes closed)

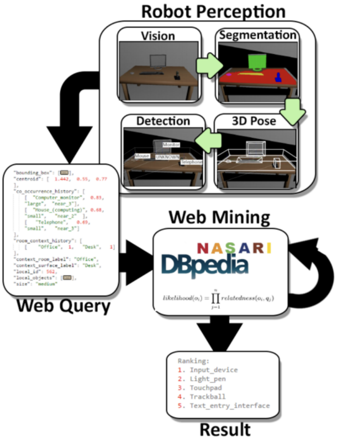
Lifelong Object Learning



J. Young et al., "Towards Lifelong Object Learning by Integrating Situated Robot Perception and Semantic Web Mining," in *Proc. 22nd European Conf. Artificial Intelligence (ECAI)*, 2016, pp. 1458–1466. Available: https://doi.org/10.3233/978-1-61499-672-9-1458

- Robotic perceptual system are often based on an unreasonable assumption that all relevant objects are known at design / robot training time

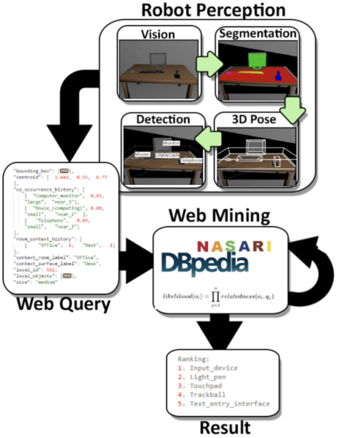
Lifelong Object Learning



J. Young et al., "Towards Lifelong Object Learning by Integrating Situated Robot Perception and Semantic Web Mining," in *Proc. 22nd European Conf. Artificial Intelligence (ECAI)*, 2016, pp. 1458–1466. Available: https://doi.org/10.3233/978-1-61499-672-9-1458

- Robotic perceptual system are often based on an unreasonable assumption that all relevant objects are known at design / robot training time
- In cognition-enabled manipulation, this assumption is relaxed by **allowing the robot to recognise unknown objects** and **endowing it with an ability to learn models of them**

Lifelong Object Learning



J. Young et al., "Towards Lifelong Object Learning by Integrating Situated Robot Perception and Semantic Web Mining," in *Proc. 22nd European Conf. Artificial Intelligence (ECAI)*, 2016, pp. 1458–1466. Available: https://doi.org/10.3233/978-1-61499-672-9-1458

- Robotic perceptual system are often based on an unreasonable assumption that all relevant objects are known at design / robot training time
- In cognition-enabled manipulation, this assumption is relaxed by **allowing the robot to recognise unknown objects** and **endowing it with an ability to learn models of them**
- Lifelong object learning can be particularly difficult in connectionist systems because of **unwanted forgetting** (it is tricky to guarantee that new information can be incorporated without forgetting relevant old information)

Class Incremental Learning

- Class incremental learning is a lifelong learning technique based on which **new classes are included in a recognition system over time**

Class Incremental Learning

- Class incremental learning is a lifelong learning technique based on which **new classes are included in a recognition system over time**
- In the context of neural networks, this would mean retraining the network with the complete (growing) dataset — not computationally feasible

Class Incremental Learning

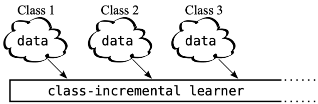
- Class incremental learning is a lifelong learning technique based on which **new classes are included in a recognition system over time**
- In the context of neural networks, this would mean retraining the network with the complete (growing) dataset — not computationally feasible
- Typical techniques use **exemplars** (representative examples of old classes) instead of keeping the full dataset

Class Incremental Learning

- Class incremental learning is a lifelong learning technique based on which **new classes are included in a recognition system over time**
- In the context of neural networks, this would mean retraining the network with the complete (growing) dataset — not computationally feasible
- Typical techniques use **exemplars** (representative examples of old classes) instead of keeping the full dataset
- A potential problem that can appear with this technique is **recency bias** (old classes get forgotten, the recognition system is biased towards recent classes) — but there are some techniques to address this problem

Class Incremental Learning

- Class incremental learning is a lifelong learning technique based on which **new classes are included in a recognition system over time**
- In the context of neural networks, this would mean retraining the network with the complete (growing) dataset — not computationally feasible
- Typical techniques use **exemplars** (representative examples of old classes) instead of keeping the full dataset
- A potential problem that can appear with this technique is **recency bias** (old classes get forgotten, the recognition system is biased towards recent classes) — but there are some techniques to address this problem



Algorithm 1 iCaRL CLASSIFY

input x // image to be classified
require $\mathcal{P} = (P_1, \ldots, P_t)$ // class exemplar sets
require $\varphi : \mathcal{X} \to \mathbb{R}^d$ // feature map
  for $y = 1, \ldots, t$ **do**
    $\mu_y \leftarrow \frac{1}{|P_y|} \sum_{p \in P_y} \varphi(p)$ // mean-of-exemplars
  end for
  $y^* \leftarrow \underset{y=1,\ldots,t}{\operatorname{argmin}} \|\varphi(x) - \mu_y\|$ // nearest prototype
output class label y^*

Algorithm 2 iCaRL INCREMENTALTRAIN

input $X^s, \ldots, X^t$ // training examples in per-class sets
input K // memory size
require Θ // current model parameters
require $\mathcal{P} = (P_1, \ldots, P_{s-1})$ // current exemplar sets
  $\Theta \leftarrow$ UPDATEREPRESENTATION$(X^s, \ldots, X^t; \mathcal{P}, \Theta)$
  $m \leftarrow K/t$ // number of exemplars per class
  for $y = 1, \ldots, s-1$ **do**
    $P_y \leftarrow$ REDUCEEXEMPLARSET(P_y, m)
  end for
  for $y = s, \ldots, t$ **do**
    $P_y \leftarrow$ CONSTRUCTEXEMPLARSET(X_y, m, Θ)
  end for
  $\mathcal{P} \leftarrow (P_1, \ldots, P_t)$ // new exemplar sets

S. Rebuffi et al., "iCaRL: Incremental Classifier and Representation Learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010. Available: https://doi.org/10.1109/CVPR.2017.587

Learning-Based Robot Manipulation

Why Learning-Based Manipulation?

A multitude of reasons — it can equip a robot with an ability to:

Improve skills based on experience

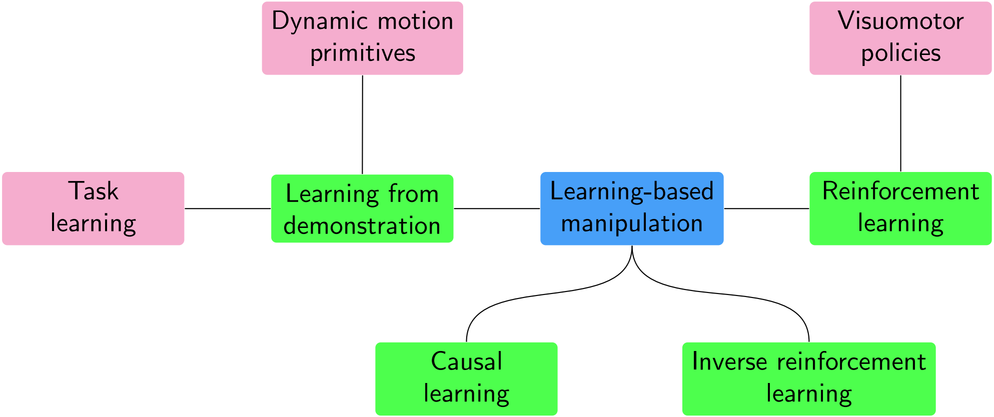
Imitate motions or acquire complete behaviour models by observing humans

Perform flexible behaviours that are difficult to program explicitly (such as visuomotor policies)

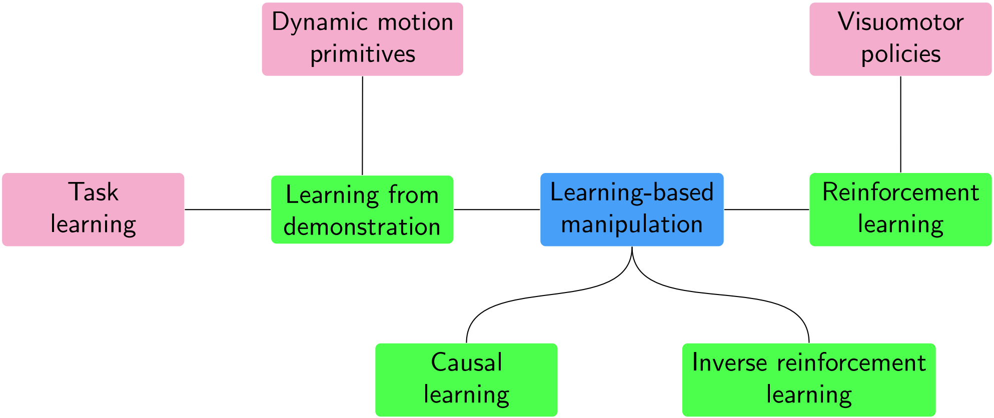
Independently explore the environment and identify causal relationships between its actions and the observed effects

Adapt based on the preferences of human collaborators

An (Incomplete) Overview of Learning-Based Manipulation



An (Incomplete) Overview of Learning-Based Manipulation



We will look at these briefly on the next slides; most of them are treated in more detail in my "Robot Learning" course

Learning from Demonstration (LfD)

- The most prevalent type of learning-based manipulation is learning from demonstration (also called imitation learning)

Learning from Demonstration (LfD)

- The most prevalent type of learning-based manipulation is learning from demonstration (also called imitation learning)
- In LfD, **an expert demonstrator teaches the robot** a motion or a complete task

Learning from Demonstration (LfD)

- The most prevalent type of learning-based manipulation is learning from demonstration (also called imitation learning)
- In LfD, **an expert demonstrator teaches the robot** a motion or a complete task
- Particularly in the context of motions, LfD is usually followed by subsequent learning (e.g. reinforcement learning); the demonstration is thus used to **initialise the robot's execution policy** so that it can then be **improved by additional self-learning**

Learning from Demonstration (LfD)

- The most prevalent type of learning-based manipulation is learning from demonstration (also called imitation learning)
- In LfD, **an expert demonstrator teaches the robot** a motion or a complete task
- Particularly in the context of motions, LfD is usually followed by subsequent learning (e.g. reinforcement learning); the demonstration is thus used to **initialise the robot's execution policy** so that it can then be **improved by additional self-learning**

H. Ravichandar et al., "Recent Advances in Robot Learning from Demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 13:1–34, 2020. Available: https://doi.org/10.1146/annurev-control-100819-063206

Dynamic Motion Primitives (DMPs)[2]

A DMP models trajectories by a second-order differential equation

$$\tau \ddot{\boldsymbol{y}} = \alpha \left(\beta \left(\boldsymbol{g} - \boldsymbol{y} \right) - \dot{\boldsymbol{y}} \right) + \boldsymbol{f}$$

Here:

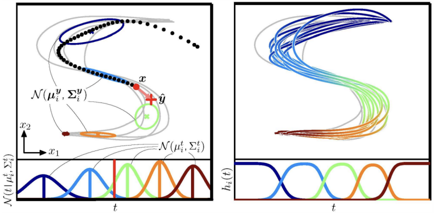
$$\boldsymbol{f}(x) = \frac{\sum_{i=1}^{k} \Psi_i(x) w_i}{\sum_{i=1}^{k} \Psi_i(x)} x \left(\boldsymbol{g} - \boldsymbol{y}_0 \right) \qquad \Psi_i(x) = \exp\left(-\frac{1}{2\sigma_i^2} (x - c_i)^2 \right)$$

By modifying the weighting terms (learned using weighted linear regression), arbitrary trajectories can be represented

[2] A. J. Ijspeert et al., "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013. Available: https://ieeexplore.ieee.org/document/6797340

Learning Trajectories Using Gaussian Mixture Models (GMMs)[3]

- Motion trajectories can also be represented in a probabilistic manner — a more appropriate model when multiple demonstrations are given. A GMM is one possible representation for this



- For a time-dependent system, the distribution $P(t, \boldsymbol{y})$ of time t and trajectory attractor points $\boldsymbol{y}$ is modelled as a GMM with K components:

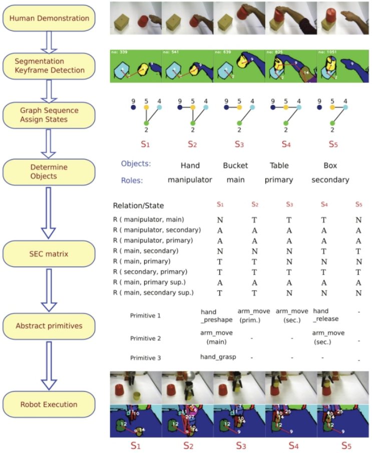
$$P(t, \boldsymbol{y}) = \sum_{i=1}^{K} \pi_i \mathcal{N}\left(\mu_i, \Sigma_i\right)$$

- The conditional distribution $P(\boldsymbol{y}|t)$ is then found by Gaussian mixture regression:

$$\hat{\mu}^{\boldsymbol{y}} = \sum_{i=1}^{K} h_i(t) \left[\mu_i^{\boldsymbol{y}} + \Sigma_i^{\boldsymbol{y}t} \left(\Sigma_i^t\right)^{-1} \left(t - \mu_i^t\right)\right] \qquad \hat{\Sigma}^{\boldsymbol{y}} = \sum_{i=1}^{K} h_i^2(t) \left[\Sigma_i^{\boldsymbol{y}} - \Sigma_i^{\boldsymbol{y}t} \left(\Sigma_i^t\right)^{-1} \Sigma_i^{t\boldsymbol{y}}\right]$$

[3] Slide fully based on S. Calinon et al., "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2012, pp. 323-329. Available: https://doi.org/10.1109/HUMANOIDS.2012.6651539

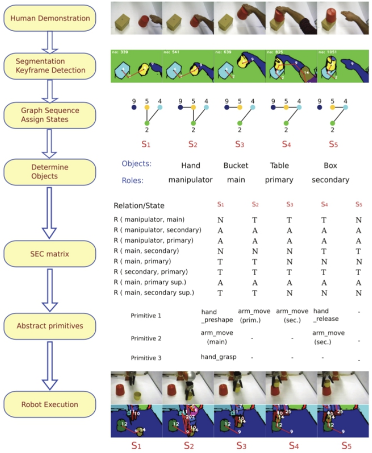
Task Learning From Observation



M. J. Aein et al., "Library of actions: Implementing a generic robot execution framework by using manipulation action semantics," *Int. Journal Robotics Research*, vol. 38, no. 8, pp. 910–934, 2019. Available: https://doi.org/10.1177/0278364919850295

- In observation-based task learning, a robot learns a complete model of a task — either for recognising or for performing the task

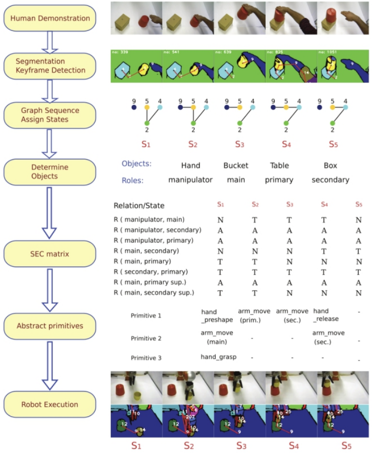
Task Learning From Observation



M. J. Aein et al., "Library of actions: Implementing a generic robot execution framework by using manipulation action semantics," *Int. Journal Robotics Research*, vol. 38, no. 8, pp. 910–934, 2019. Available: https://doi.org/10.1177/0278364919850295

- In observation-based task learning, a robot learns a complete model of a task — either for recognising or for performing the task
- This is **particularly useful for recognition if a robot can learn from a large video dataset** — demonstrations are then given "for free"

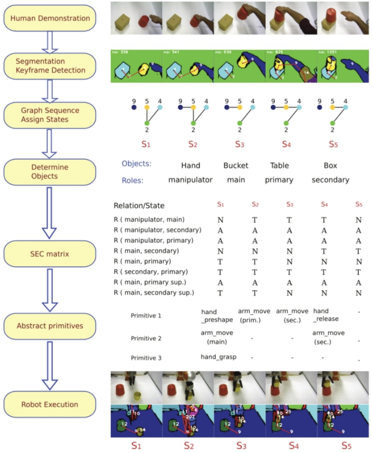
Task Learning From Observation



M. J. Aein et al., "Library of actions: Implementing a generic robot execution framework by using manipulation action semantics," *Int. Journal Robotics Research*, vol. 38, no. 8, pp. 910–934, 2019. Available: https://doi.org/10.1177/0278364919850295

- In observation-based task learning, a robot learns a complete model of a task — either for recognising or for performing the task
- This is **particularly useful for recognition if a robot can learn from a large video dataset** — demonstrations are then given "for free"
- **Learning models for execution usually requires data other than images** (e.g. skeleton recordings)

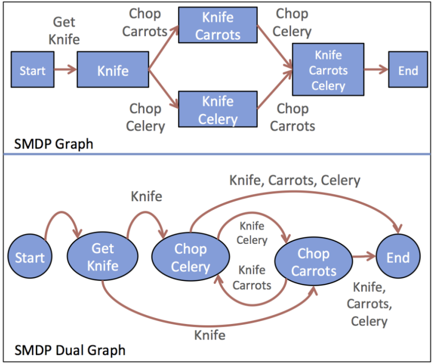
Task Learning From Observation



M. J. Aein et al., "Library of actions: Implementing a generic robot execution framework by using manipulation action semantics," *Int. Journal Robotics Research*, vol. 38, no. 8, pp. 910–934, 2019. Available: https://doi.org/10.1177/0278364919850295

- In observation-based task learning, a robot learns a complete model of a task — either for recognising or for performing the task
- This is **particularly useful for recognition if a robot can learn from a large video dataset** — demonstrations are then given "for free"
- **Learning models for execution usually requires data other than images** (e.g. skeleton recordings)
- Typically used in cognitivist and hybrid systems — the learning process involves grounding symbols of observed entities

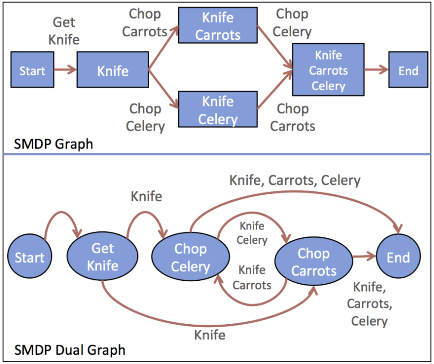
Active Manipulation Task Learning



B. Hayes and B. Scassellati, "Discovering task constraints through observation and active learning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 4442–4449. Available: https://doi.org/10.1109/IROS.2014.6943191

- During task learning, a robot doesn't have to be a passive observer — it can also **actively participate in the learning process by posing disambiguating queries**

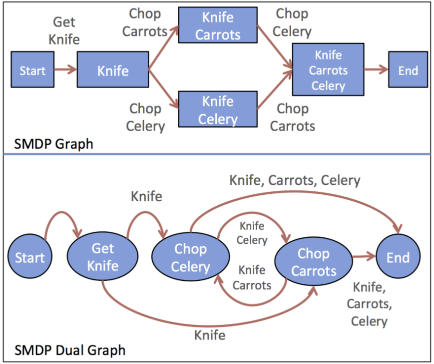
Active Manipulation Task Learning



B. Hayes and B. Scassellati, "Discovering task constraints through observation and active learning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 4442–4449. Available: https://doi.org/10.1109/IROS.2014.6943191

- During task learning, a robot doesn't have to be a passive observer — it can also **actively participate in the learning process by posing disambiguating queries**
- This is particularly important if multiple distinct demonstrations (from multiple teachers) are received — actively asking queries will allow the robot to identify actions:

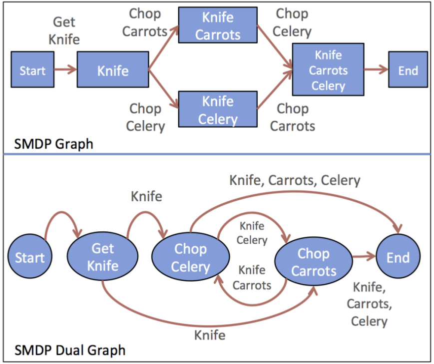
Active Manipulation Task Learning



B. Hayes and B. Scassellati, "Discovering task constraints through observation and active learning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 4442–4449. Available: https://doi.org/10.1109/IROS.2014.6943191

- During task learning, a robot doesn't have to be a passive observer — it can also **actively participate in the learning process by posing disambiguating queries**
- This is particularly important if multiple distinct demonstrations (from multiple teachers) are received — actively asking queries will allow the robot to identify actions:
  - **whose ordering is not strictly important**

Active Manipulation Task Learning



B. Hayes and B. Scassellati, "Discovering task constraints through observation and active learning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2014, pp. 4442–4449. Available: https://doi.org/10.1109/IROS.2014.6943191

- During task learning, a robot doesn't have to be a passive observer — it can also **actively participate in the learning process by posing disambiguating queries**
- This is particularly important if multiple distinct demonstrations (from multiple teachers) are received — actively asking queries will allow the robot to identify actions:
  - **whose ordering is not strictly important** or
  - **which are optional in the task**

Visuomotor Policies

- An **execution policy that continuously maps perceptual input to motor actions** is called a visuomotor policy

Visuomotor Policies

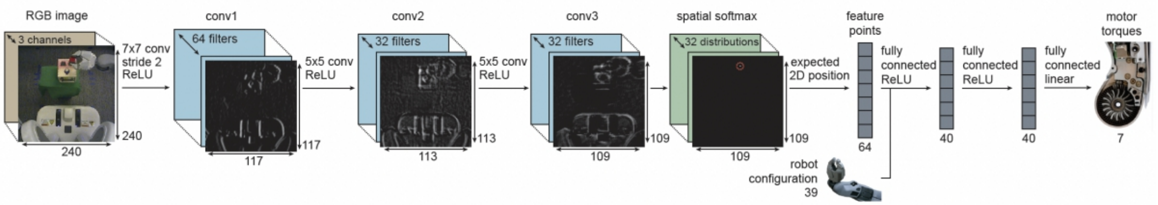
- An **execution policy that continuously maps perceptual input to motor actions** is called a visuomotor policy
- As the name suggests, **the perceptual input needs to include visual data**, although other modalities are sometimes combined as well

Visuomotor Policies

- An **execution policy that continuously maps perceptual input to motor actions** is called a visuomotor policy
- As the name suggests, **the perceptual input needs to include visual data**, although other modalities are sometimes combined as well
- Conceptually, **visuomotor policies mimic the way people manipulate** — our manipulation actions are always driven by perceptual input and we don't need to always replan trajectories

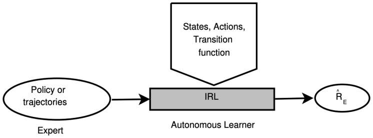
Visuomotor Policies

- An **execution policy that continuously maps perceptual input to motor actions** is called a visuomotor policy
- As the name suggests, **the perceptual input needs to include visual data**, although other modalities are sometimes combined as well
- Conceptually, **visuomotor policies mimic the way people manipulate** — our manipulation actions are always driven by perceptual input and we don't need to always replan trajectories
- Advances in deep learning have made visuomotor policies practically feasible; such policies are often trained using deep reinforcement learning



S. Levine et al., "End-to-End Training of Deep Visuomotor Policies," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, Jan. 2016. Available: https://jmlr.org/papers/v17/15-522.html

Inverse Reinforcement Learning (IRL)



Algorithm 1: Template for IRL

Input: $\mathcal{M}\backslash_{R_E} = \langle S, A, T, \gamma \rangle$,
Set of trajectories demonstrating desired behavior:
$\mathcal{D} = \{\langle (s_0, a_0), (s_1, a_1), \ldots, (s_t, a_t) \rangle, \ldots\}$, $s_t \in S$, $a_t \in A$, $t \in \mathbb{N}$,
or expert's policy: π_E, and reward function features

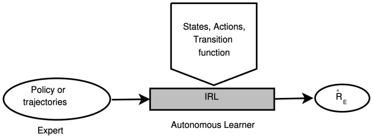
Output: $\hat{R}_E$

1 Model the expert's observed behavior as the solution of an MDP whose reward function is not known;
2 Initialize the parameterized form of the reward function using any given features (linearly weighted sum of feature values, distribution over rewards, or other);
3 Solve the MDP with current reward function to generate the learned behavior or policy;
4 Update the optimization parameters to minimize the divergence between the observed behavior (or policy) and the learned behavior (policy);
5 Repeat the previous two steps till the divergence is reduced to a desired level.

S. Arora and P. Yoshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, pp. 103500:1-28, Aug. 2021. Available: https://doi.org/10.1016/j.artint.2021.103500

- In robot (reinforcement) learning, we assume that a reward function to learn from is given —— that is, however, not always the case

Inverse Reinforcement Learning (IRL)



Algorithm 1: Template for IRL

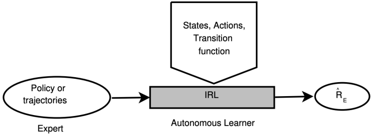
Input: $\mathcal{M}\backslash_{R_E} = \langle S, A, T, \gamma \rangle$,
Set of trajectories demonstrating desired behavior:
$\mathcal{D} = \{\langle (s_0, a_0), (s_1, a_1), \ldots, (s_t, a_t)\rangle, \ldots\}$, $s_t \in S$, $a_t \in A$, $t \in \mathbb{N}$,
or expert's policy: π_E, and reward function features
Output: $\hat{R}_E$

1 Model the expert's observed behavior as the solution of an MDP whose reward function is not known;
2 Initialize the parameterized form of the reward function using any given features (linearly weighted sum of feature values, distribution over rewards, or other);
3 Solve the MDP with current reward function to generate the learned behavior or policy;
4 Update the optimization parameters to minimize the divergence between the observed behavior (or policy) and the learned behavior (policy);
5 Repeat the previous two steps till the divergence is reduced to a desired level.

S. Arora and P. Yoshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, pp. 103500:1-28, Aug. 2021. Available: https://doi.org/10.1016/j.artint.2021.103500

- In robot (reinforcement) learning, we assume that a reward function to learn from is given —— that is, however, not always the case
- The goal of IRL is to **estimate the reward function** that a human expert is following

Inverse Reinforcement Learning (IRL)



Algorithm 1: Template for IRL

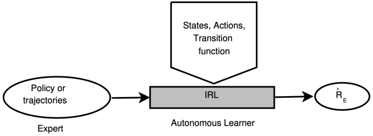
Input: $\mathcal{M}\backslash_{R_E} = \langle S, A, T, \gamma \rangle$,
Set of trajectories demonstrating desired behavior:
$\mathcal{D} = \{\langle (s_0, a_0), (s_1, a_1), \ldots, (s_t, a_t) \rangle, \ldots\}$, $s_t \in S$, $a_t \in A$, $t \in \mathbb{N}$,
or expert's policy: π_E, and reward function features
Output: $\hat{R}_E$

1 Model the expert's observed behavior as the solution of an MDP whose reward function is not known;
2 Initialize the parameterized form of the reward function using any given features (linearly weighted sum of feature values, distribution over rewards, or other);
3 Solve the MDP with current reward function to generate the learned behavior or policy;
4 Update the optimization parameters to minimize the divergence between the observed behavior (or policy) and the learned behavior (policy);
5 Repeat the previous two steps till the divergence is reduced to a desired level.

S. Arora and P. Yoshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, pp. 103500:1-28, Aug. 2021. Available: https://doi.org/10.1016/j.artint.2021.103500

- In robot (reinforcement) learning, we assume that a reward function to learn from is given — that is, however, not always the case
- The goal of IRL is to **estimate the reward function** that a human expert is following
- A typical IRL learning loop involves (i) **finding a reward estimate** and (ii) **learning a policy with the estimate** until the learned policy corresponds to the observed policy

Inverse Reinforcement Learning (IRL)



Algorithm 1: Template for IRL

Input: $\mathcal{M}\backslash_{R_E} = \langle S, A, T, \gamma \rangle$,
Set of trajectories demonstrating desired behavior:
$\mathcal{D} = \{\langle (s_0, a_0), (s_1, a_1), \ldots, (s_t, a_t) \rangle, \ldots\}$, $s_t \in S$, $a_t \in A$, $t \in \mathbb{N}$,
or expert's policy: π_E, and reward function features
Output: $\hat{R}_E$

1 Model the expert's observed behavior as the solution of an MDP whose reward function is not known;
2 Initialize the parameterized form of the reward function using any given features (linearly weighted sum of feature values, distribution over rewards, or other);
3 Solve the MDP with current reward function to generate the learned behavior or policy;
4 Update the optimization parameters to minimize the divergence between the observed behavior (or policy) and the learned behavior (policy);
5 Repeat the previous two steps till the divergence is reduced to a desired level.

S. Arora and P. Yoshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, pp. 103500:1-28, Aug. 2021. Available: https://doi.org/10.1016/j.artint.2021.103500

- In robot (reinforcement) learning, we assume that a reward function to learn from is given — that is, however, not always the case
- The goal of IRL is to **estimate the reward function** that a human expert is following
- A typical IRL learning loop involves (i) **finding a reward estimate** and (ii) **learning a policy with the estimate** until the learned policy corresponds to the observed policy
- Conceptually, IRL corresponds very closely to how human apprentices learn by observing experts

Causal Learning

- **Typical machine learning leads to learning correlations, not necessarily causations** (for example, this is one reason why deep learning-based models can be vulnerable to trivial adversarial attacks)

Causal Learning

- **Typical machine learning leads to learning correlations, not necessarily causations** (for example, this is one reason why deep learning-based models can be vulnerable to trivial adversarial attacks)
- Causal learning can have a variety of purposes, such as:

Causal Learning

- **Typical machine learning leads to learning correlations, not necessarily causations** (for example, this is one reason why deep learning-based models can be vulnerable to trivial adversarial attacks)
- Causal learning can have a variety of purposes, such as:
  - Finding out **which parameters are actually relevant for learning a given skill** (from a given set of parameters, not all of which may be important for the skill)

Causal Learning

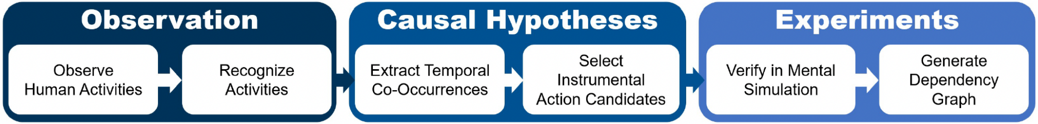
- **Typical machine learning leads to learning correlations, not necessarily causations** (for example, this is one reason why deep learning-based models can be vulnerable to trivial adversarial attacks)
- Causal learning can have a variety of purposes, such as:
  - Finding out **which parameters are actually relevant for learning a given skill** (from a given set of parameters, not all of which may be important for the skill)
  - **Identifying dependences between skills** (information that is useful for fault analysis)

Causal Learning

- **Typical machine learning leads to learning correlations, not necessarily causations** (for example, this is one reason why deep learning-based models can be vulnerable to trivial adversarial attacks)
- Causal learning can have a variety of purposes, such as:
  - Finding out **which parameters are actually relevant for learning a given skill** (from a given set of parameters, not all of which may be important for the skill)
  - **Identifying dependences between skills** (information that is useful for fault analysis)
- Successful causal learning requires the ability to perform **interventions** (either in the real world or in a simulated model) — **identifying causality requires different hypotheses to be tested and incorrect ones to be eliminated**

Causal Learning

- **Typical machine learning leads to learning correlations, not necessarily causations** (for example, this is one reason why deep learning-based models can be vulnerable to trivial adversarial attacks)
- Causal learning can have a variety of purposes, such as:
  - Finding out **which parameters are actually relevant for learning a given skill** (from a given set of parameters, not all of which may be important for the skill)
  - **Identifying dependences between skills** (information that is useful for fault analysis)
- Successful causal learning requires the ability to perform **interventions** (either in the real world or in a simulated model) — **identifying causality requires different hypotheses to be tested and incorrect ones to be eliminated**



C. Uhde et al., "The Robot as Scientist: Using Mental Simulation to Test Causal Hypotheses Extracted from Human Activities in Virtual Reality," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2020, pp. 8081–8086. Available: https://doi.org/10.1109/IROS45743.2020.9341505

Knowledge for Manipulation

Useful Types of Knowledge for Cognition-Enabled Manipulation

Different types of knowledge can facilitate cognition-enabled manipulation:

Object ontologies

Specify properties of objects that a robot needs to manipulate, or relations and similarities between objects

World models

Represent the robot's up-to-date belief about the state of the world — the objects and other agents in it

Knowledge sharing

Allow knowledge reuse among agents

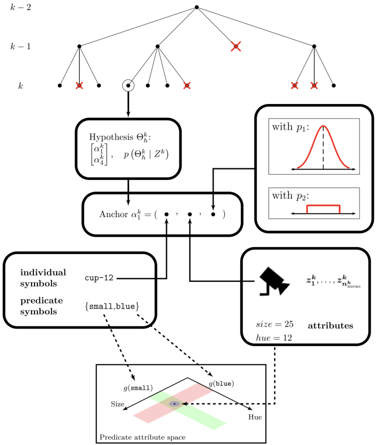
Predictive skill models

Enable joint task and motion planning

Prior learned experiences

Facilitate subsequent learning and knowledge transfer

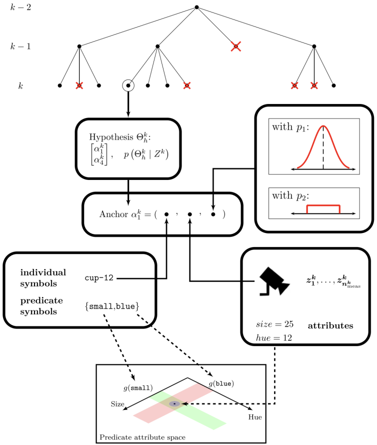
World Modelling



J. Elfring et al., "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Feb. 2013. Available: https://doi.org/10.1016/j.robot.2012.11.005

- A world model is a **representation that specifies the robot's belief about various relevant properties of the world**

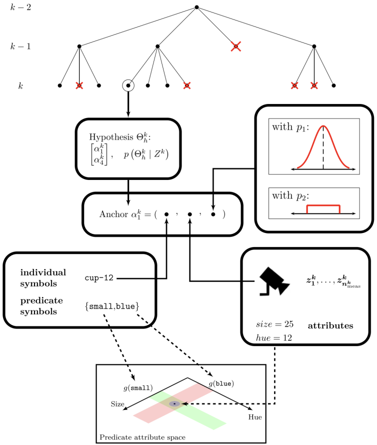
World Modelling



J. Elfring et al., "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Feb. 2013. Available: https://doi.org/10.1016/j.robot.2012.11.005

- A world model is a **representation that specifies the robot's belief about various relevant properties of the world**
- A world model enables a robot to:

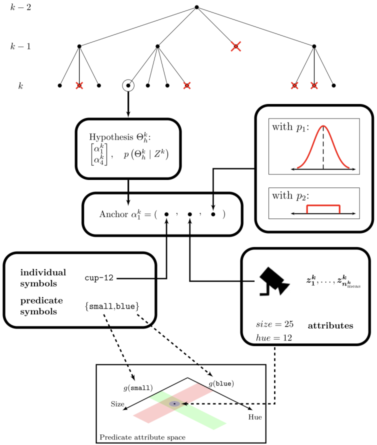
World Modelling



J. Elfring et al., "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Feb. 2013. Available: https://doi.org/10.1016/j.robot.2012.11.005

- A world model is a **representation that specifies the robot's belief about various relevant properties of the world**
- A world model enables a robot to:
  - **Keep track of dynamic changes** in the world

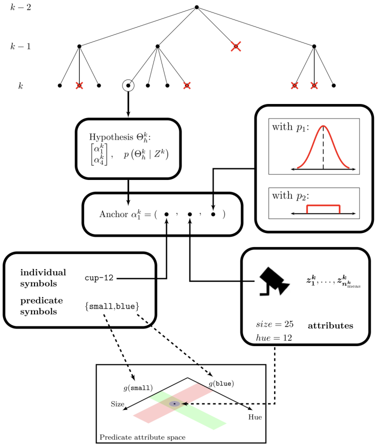
World Modelling



J. Elfring et al., "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Feb. 2013. Available: https://doi.org/10.1016/j.robot.2012.11.005

- A world model is a **representation that specifies the robot's belief about various relevant properties of the world**
- A world model enables a robot to:
  - **Keep track of dynamic changes** in the world
  - **Actively explore hypotheses in order to improve its belief**

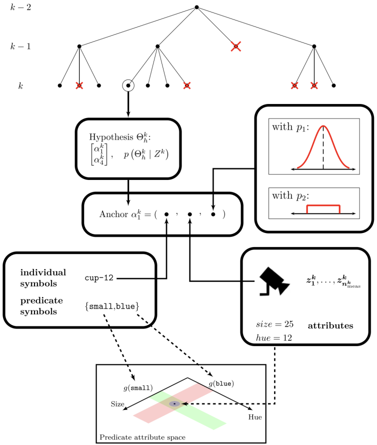
World Modelling



J. Elfring et al., "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Feb. 2013. Available: https://doi.org/10.1016/j.robot.2012.11.005

- A world model is a **representation that specifies the robot's belief about various relevant properties of the world**
- A world model enables a robot to:
  - **Keep track of dynamic changes** in the world
  - **Actively explore hypotheses in order to improve its belief**
  - **Communicate its belief to other agents** during collaborative execution

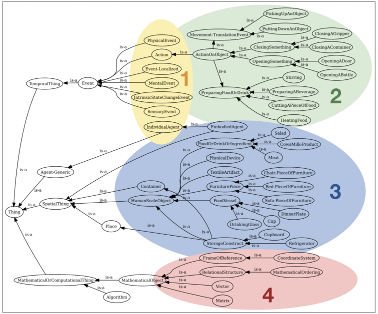
World Modelling



J. Elfring et al., "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Feb. 2013. Available: https://doi.org/10.1016/j.robot.2012.11.005

- A world model is a **representation that specifies the robot's belief about various relevant properties of the world**
- A world model enables a robot to:
  - **Keep track of dynamic changes** in the world
  - **Actively explore hypotheses in order to improve its belief**
  - **Communicate its belief to other agents** during collaborative execution
- Different representations used in cognitivist and emergent systems

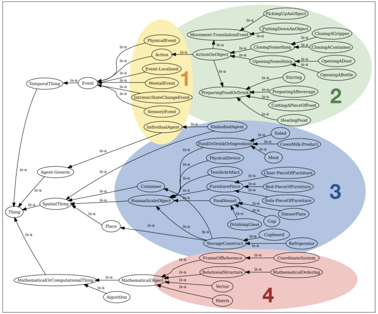
Object Ontologies



M. Tenorth and M. Beetz, "KnowRob: A knowledge processing infrastructure for cognitionenabled robots," *Int. Journal Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013. Available: https://doi.org/10.1177/0278364913481635

- An ontology allows specifying information about an environment using **description logic**

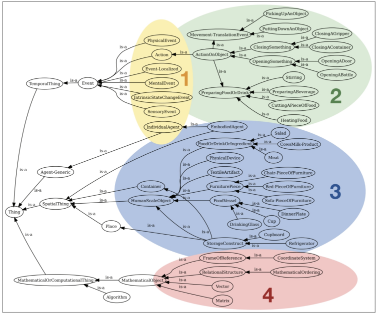
Object Ontologies



M. Tenorth and M. Beetz, "KnowRob: A knowledge processing infrastructure for cognitionenabled robots," *Int. Journal Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013. Available: https://doi.org/10.1177/0278364913481635

- An ontology allows specifying information about an environment using **description logic**
- An ontology consist of two components:
  - A **terminological box (TBox)**, which includes concepts (e.g. objects) and relations between them
  - An **assertional box (ABox)**, which includes instantiations of the concepts and relations

Object Ontologies



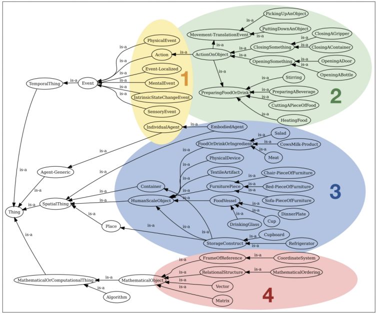
M. Tenorth and M. Beetz, "KnowRob: A knowledge processing infrastructure for cognitionenabled robots," *Int. Journal Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013. Available: https://doi.org/10.1177/0278364913481635

- An ontology allows specifying information about an environment using **description logic**

- An ontology consist of two components:
  - A **terminological box (TBox)**, which includes concepts (e.g. objects) and relations between them
  - An **assertional box (ABox)**, which includes instantiations of the concepts and relations

- We generally distinguish between two types of ontologies:
  - An **upper-level ontology** that specifies common concepts and relations that hold across domains
  - A **domain-specific ontology** that instantiates an upper-level ontology or extends it with terms that are specific to a given domain

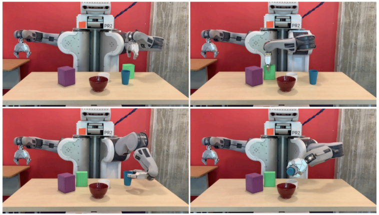
Object Ontologies



M. Tenorth and M. Beetz, "KnowRob: A knowledge processing infrastructure for cognitionenabled robots," *Int. Journal Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013. Available: https://doi.org/10.1177/0278364913481635

- An ontology allows specifying information about an environment using **description logic**
- An ontology consist of two components:
  - A **terminological box (TBox)**, which includes concepts (e.g. objects) and relations between them
  - An **assertional box (ABox)**, which includes instantiations of the concepts and relations
- We generally distinguish between two types of ontologies:
  - An **upper-level ontology** that specifies common concepts and relations that hold across domains
  - A **domain-specific ontology** that instantiates an upper-level ontology or extends it with terms that are specific to a given domain
- Only used in cognitivist and hybrid cognitive systems

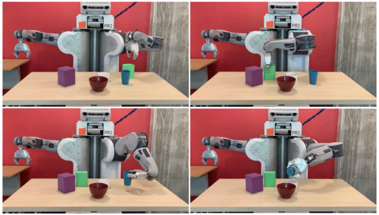
Task and Motion Planning (TAMP)



C. R. Garrett et al. "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021. Available: https://doi.org/10.1146/annurev-control-091420-084139

- Task and motion planning is the act of performing **simultaneous task planning and robot motion planning** before a robot starts with the execution of a task

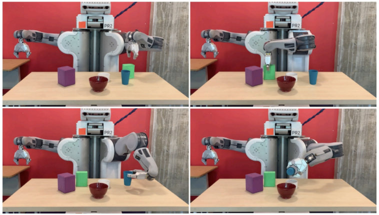
Task and Motion Planning (TAMP)



C. R. Garrett et al. "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021. Available: https://doi.org/10.1146/annurev-control-091420-084139

- Task and motion planning is the act of performing **simultaneous task planning and robot motion planning** before a robot starts with the execution of a task
- TAMP requires **not only planning models of actions** (in terms of preconditions and effects), **but also predictive execution models** so that the most appropriate execution parameters can be identified

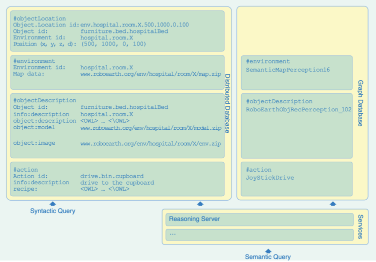
Task and Motion Planning (TAMP)



C. R. Garrett et al. “Integrated task and motion planning,” *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021. Available: https://doi.org/10.1146/annurev-control-091420-084139

- Task and motion planning is the act of performing **simultaneous task planning and robot motion planning** before a robot starts with the execution of a task
- TAMP requires **not only planning models of actions** (in terms of preconditions and effects), **but also predictive execution models** so that the most appropriate execution parameters can be identified
- Particularly useful if the robot's operation is not interrupted by other agents

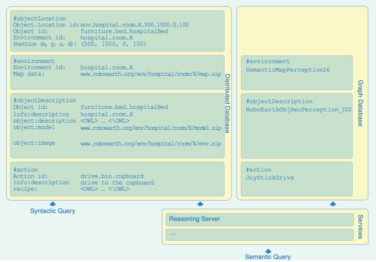
Knowledge Sharing Between Robots



M. Waibel et al., "RoboEarth," in *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, June 2011. Available: https://doi.org/10.1109/MRA.2011.941632

- Cognitive manipulation can also be facilitated by **enabling robots to reuse knowledge that other robots possess**

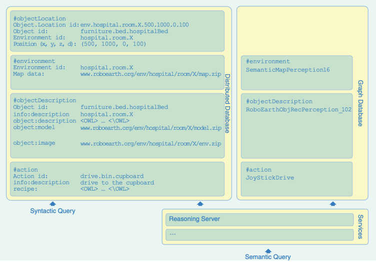
Knowledge Sharing Between Robots



M. Waibel et al., "RoboEarth," in *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, June 2011. Available: https://doi.org/10.1109/MRA.2011.941632

- Cognitive manipulation can also be facilitated by **enabling robots to reuse knowledge that other robots possess**
- Such knowledge exchange can be done either between **members of a multi-robot team** (using network communication) or even between **completely distinct robots** (through web-based services)

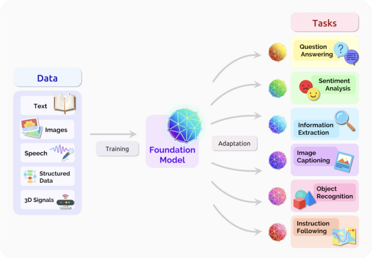
Knowledge Sharing Between Robots



M. Waibel et al., "RoboEarth," in *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, June 2011. Available: https://doi.org/10.1109/MRA.2011.941632

- Cognitive manipulation can also be facilitated by **enabling robots to reuse knowledge that other robots possess**
- Such knowledge exchange can be done either between **members of a multi-robot team** (using network communication) or even between **completely distinct robots** (through web-based services)
- Successful knowledge sharing requires:
  - **A common knowledge representation framework** (e.g. a knowledge graph)
  - **A unified querying language for knowledge retrieval**

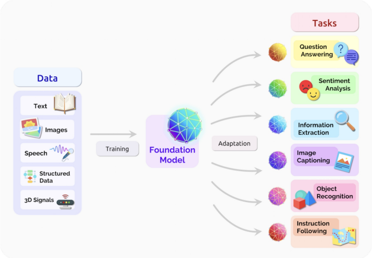
Prior Learned Experiences: Foundation Models



R. Bommasani et al. "On the opportunities and risks of foundation models," *CoRR*, vol. 2108.07258, 2021. Available: https://arxiv.org/abs/2108.07258

- The training of large data-driven models for different tasks leads to a question of **how the knowledge in such models can be transferred to different tasks**

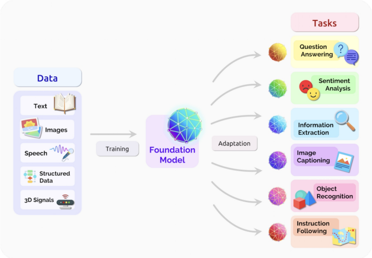
Prior Learned Experiences: Foundation Models



R. Bommasani et al. "On the opportunities and risks of foundation models," *CoRR*, vol. 2108.07258, 2021. Available: https://arxiv.org/abs/2108.07258

- The training of large data-driven models for different tasks leads to a question of **how the knowledge in such models can be transferred to different tasks**
- A foundation model is a **large neural network model that is trained on a large dataset** (potentially including different modalities); this is then **adapted (fine-tuned) for a concrete task** (with task-specific data)

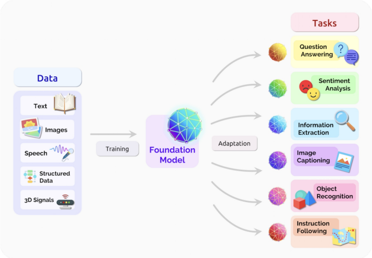
Prior Learned Experiences: Foundation Models



R. Bommasani et al. "On the opportunities and risks of foundation models," *CoRR*, vol. 2108.07258, 2021. Available: https://arxiv.org/abs/2108.07258

- The training of large data-driven models for different tasks leads to a question of **how the knowledge in such models can be transferred to different tasks**
- A foundation model is a **large neural network model that is trained on a large dataset** (potentially including different modalities); this is then **adapted (fine-tuned) for a concrete task** (with task-specific data)
- Foundation models can be used in emergent and hybrid cognitive architectures

Prior Learned Experiences: Foundation Models



R. Bommasani et al. "On the opportunities and risks of foundation models," *CoRR*, vol. 2108.07258, 2021. Available: https://arxiv.org/abs/2108.07258

- The training of large data-driven models for different tasks leads to a question of **how the knowledge in such models can be transferred to different tasks**
- A foundation model is a **large neural network model that is trained on a large dataset** (potentially including different modalities); this is then **adapted (fine-tuned) for a concrete task** (with task-specific data)
- Foundation models can be used in emergent and hybrid cognitive architectures
- Foundation models are very large networks (which makes them inherently non-transparent — by themselves, not ideal for trustworthy robots)

Execution Monitoring and Failure Recovery

Execution Anomaly Detection

- A cognition-enabled manipulation system needs to be able to **recognise cases in which the execution does not proceed as expected**

Execution Anomaly Detection

- A cognition-enabled manipulation system needs to be able to **recognise cases in which the execution does not proceed as expected**
- Anomaly detection is a **procedure of identifying time points at which such unexpected events occur**

Execution Anomaly Detection

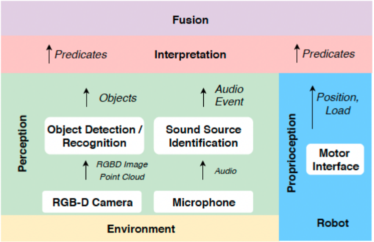
- A cognition-enabled manipulation system needs to be able to **recognise cases in which the execution does not proceed as expected**
- Anomaly detection is a **procedure of identifying time points at which such unexpected events occur**
- Note that **anomalies and failures are not necessarily equivalent**: a failure is an anomaly, but the opposite does not necessarily hold

Execution Anomaly Detection

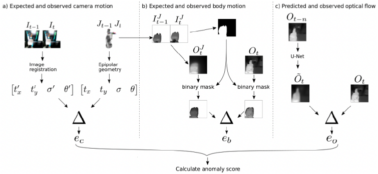
- A cognition-enabled manipulation system needs to be able to **recognise cases in which the execution does not proceed as expected**
- Anomaly detection is a **procedure of identifying time points at which such unexpected events occur**
- Note that **anomalies and failures are not necessarily equivalent**: a failure is an anomaly, but the opposite does not necessarily hold
- Anomaly (or failure detection) is a prerequisite for subsequent execution recovery

Execution Anomaly Detection

- A cognition-enabled manipulation system needs to be able to **recognise cases in which the execution does not proceed as expected**
- Anomaly detection is a **procedure of identifying time points at which such unexpected events occur**
- Note that **anomalies and failures are not necessarily equivalent**: a failure is an anomaly, but the opposite does not necessarily hold
- Anomaly (or failure detection) is a prerequisite for subsequent execution recovery



A. Inceoglu et al., "Failure Detection Using Proprioceptive, Auditory and Visual Modalities," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2018, pp. 2491–2496. Available: https://doi.org/10.1109/IROS.2018.8594169



S. Thoduka et al., "Using Visual Anomaly Detection for Task Execution Monitoring," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2021, pp. 4604–4610. Available: https://doi.org/10.1109/IROS51168.2021.9636133

Execution Failure Recovery

- Anomaly detection is useful to the execution of manipulation skills if it is followed by a subsequent recovery

Execution Failure Recovery

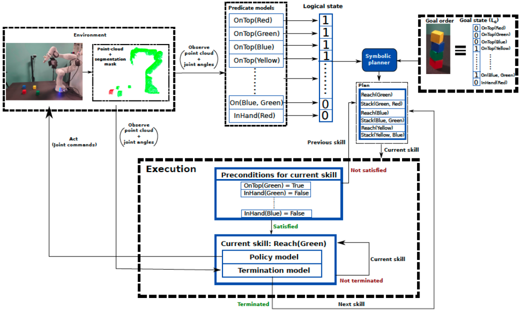
- Anomaly detection is useful to the execution of manipulation skills if it is followed by a subsequent recovery
- Execution failure recovery is a **procedure of performing actions that remedy the execution and enable the original execution to be continued**

Execution Failure Recovery

- Anomaly detection is useful to the execution of manipulation skills if it is followed by a subsequent recovery
- Execution failure recovery is a **procedure of performing actions that remedy the execution and enable the original execution to be continued**
- Often, recovery is done by a simple heuristic (e.g. repeating the failed skill); **a cognitive robot should be able to recover more effectively by identifying and using information about the failure**

Execution Failure Recovery

- Anomaly detection is useful to the execution of manipulation skills if it is followed by a subsequent recovery
- Execution failure recovery is a **procedure of performing actions that remedy the execution and enable the original execution to be continued**
- Often, recovery is done by a simple heuristic (e.g. repeating the failed skill); **a cognitive robot should be able to recover more effectively by identifying and using information about the failure**



S. Mukherjee et al., "Reactive Long Horizon Task Execution via Visual Skill and Precondition Models," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2021, pp. 5717–5724. Available: https://doi.org/10.1109/IROS51168.2021.9636037

Conclusion: What is Cognition-Enabled Robot Manipulation?

- **Cognition-enabled manipulation enhances traditional robot manipulation with capabilities associated with cognitive agents**, such as:
  - the ability to use knowledge about the environment
  - deal with varied perceptual signals
  - learn from experience
  - identify and resolve execution failures
- Through its various elements, cognition-enabled manipulation **increases the practical capabilities of a robot** and **enhances the suitability of manipulation-based systems for everyday, human-centred environments**