Commonsense Reasoning and Knowledge Acquisition to Guide Deep Learning on Robots

Robust AI for Neurorobotics Workshop

Tiago Mota [1], Heather Riley [1], Mohan Sridharan[1] [2]

[1]Department of Electrical and Computer Engineering
The University of Auckland, NZ

[2]Intelligent Robotics Lab, School of Computer Science
University of Birmingham, UK

August 27, 2019

[1]Ben Meadows, Rocio Gomez (Univ. of Auckland, NZ); Michael Gelfond (Texas Tech, USA); Jeremy Wyatt, Michael Mathew, Saif Sidhik (Univ. of Birmingham, UK); Shiqi Zhang (SUNY Binghamton, USA); Pat Langley (ISLE, USA), ONR Awards N00014-13-1-0766, N00014-17-1-2434; AFOSR Award FA2386-16-1-4071; EPSRC/EU projects

Architecture Desiderata

- Uses different descriptions of incomplete commonsense knowledge and uncertainty, and different reasoning schemes to improve decision making.
  "The cereal box is usually in the kitchen"
  "I am 90% certain the cereal box is in the kitchen"

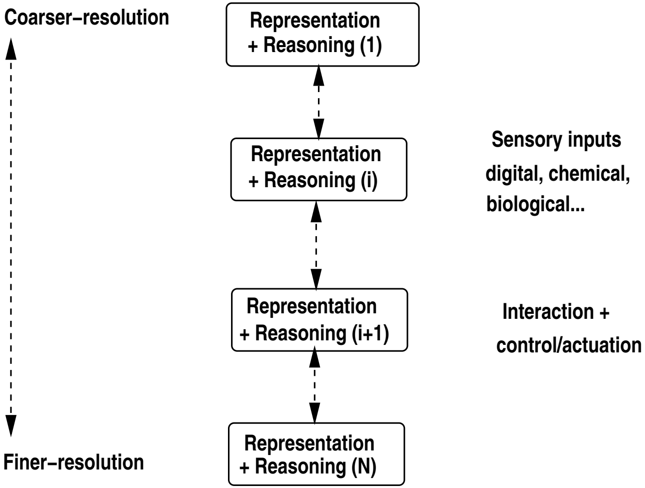
- Acquires domain knowledge, e.g., action preconditions, effects and affordances, interactively and from data.
  "A brittle object breaks when it is put down"
  "Robot with weak arm cannot lift heavy box"

- Enables designer to understand robot's behavior and establish that it satisfies desirable properties.

Inspiration and Core Ideas

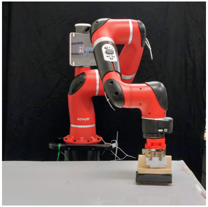
- Cognitive systems, theories of human cognition and control.
- Computational models of intention, affordance, explanation.
- Represent, reason, learn jointly at different abstractions with different schemes (Alan Turing, 1952; morphogenesis).
- Logician, statistician, and creative explorer; tight coupling not unified representation (Immanuel Kant, Aaron Sloman).
- Interactive and cumulative learning of relevant concepts.
- *Not focusing on hardware, energy requirements.*

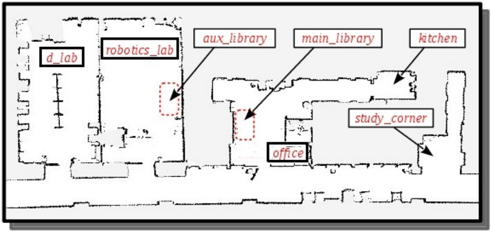
Overall Architecture: Basic Idea



Illustrative Domain: Robot Assistant

Robot assistant finding and manipulating objects.





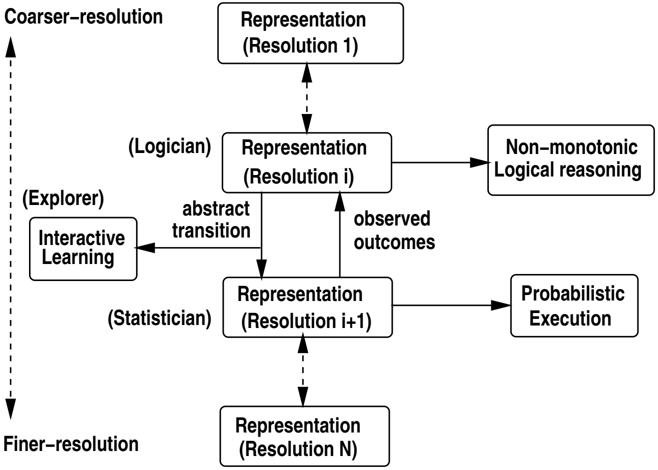
Claims: Representation

1. Distributed representation of knowledge (commonsense, probabilistic) at different abstractions.
2. Knowledge structures include definitions, constraints (static, causal/dynamic).
3. Beliefs include prior knowledge, inferences, plans, explanations.
4. History includes observations, (attempted, executed) actions.
5. Separation of concerns (domain-specific/independent knowledge, observations), but abstractions tightly coupled.
6. Possible worlds, each a set of beliefs.

Claims: Reasoning

1. Knowledge elements support non-monotonic revision; revise previously held conclusions.
2. Actions produce immediate or delayed outcomes; reward-based and architecture-based exploration.
3. Observations obtained through active exploration or reactive action execution.
4. "Here and there" reasoning; satisfiability, stochastic policies.

Refinement-Based Architecture



Action Language $\mathcal{AL}_d$

- Formal models of parts of natural language used for describing transition diagrams.
- Hierarchy of basic sorts, statics, fluents and actions.
- Types of statements:
  - Causal law (deterministic, non-deterministic).
  - State constraint and definitions.
  - Executability condition.

Coarse-Resolution Representation

- Collection of statements of $\mathcal{AL}_d$ forms system description $\mathcal{D}_c$, includes sorted signature Σ_c and axioms.

- Statics: $next\_to(place, place)$
- Fluents: $loc : robot \rightarrow place$, $in\_hand : robot \times object \rightarrow bool$

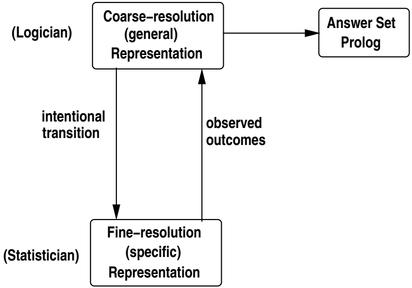
- Actions: $move(robot, place)$, $pickup(robot, object)$, $putdown(robot, object)$.

- Axioms:

$$move(rob_1, Pl) \ \textbf{causes} \ loc(rob_1) = Pl$$
$$loc(O) = Pl \ \textbf{if} \ loc(rob_1) = Pl, \ in\_hand(rob_1, O)$$
$$\textbf{impossible} \ pickup(rob_1, O) \ \textbf{if} \ loc(rob_1) \neq loc(O)$$

Coarse-Resolution History and Reasoning

- History $\mathcal{H}_c$ with observations, actions, initial state defaults.
- Logician's task:
  - **Input**: (a) $\mathcal{D}_c$; (b) $\mathcal{H}_c$; (c) Goal.
  - **Output**: diagnose, plan, next transition $T = \langle \sigma_1, a^c, \sigma_2 \rangle$.
  - Can translate to different formalisms.

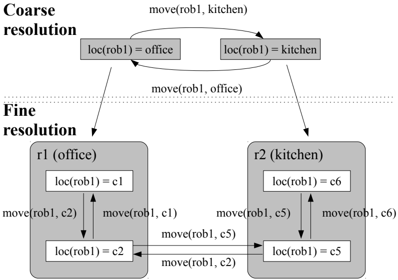


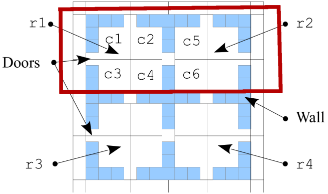
Non-monotonic Logical Reasoning

- Nonmonotonic logical reasoning with program $\Pi(\mathcal{D}_c, \mathcal{H}_c)$.
- Answer Set Prolog; reasoning by computing answer sets.
- Default negation and epistemic disjunction.

`¬ l`	l is believed to be false
`not l`	it is not believed that l is true
`p ∨ ¬ p`	is a tautology
`p or ¬ p`	is not tautological

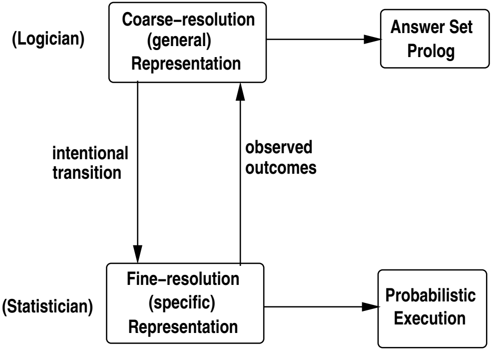
Refine + Zoom + Randomize





- Refinement: describe ($\mathcal{D}_c$) at finer resolution ($\mathcal{D}_f$).
- Theory of observation: knowledge fluents + actions.
- Randomize and zoom to $\mathcal{D}_{fr}(T)$ for $T = \langle \sigma_1, a^c, \sigma_2 \rangle$.
- Formal relationships; domain-specific knowledge.

Probabilistic Reasoning

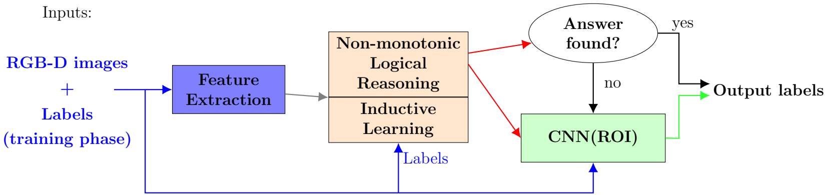


- Plan and execute probabilistically; existing algorithms (motion planners, POMDPs).
- Infer coarse-resolution outcomes from fine-resolution; add to $\mathcal{H}_c$ for subsequent reasoning.

Reasoning + Learning: VQA

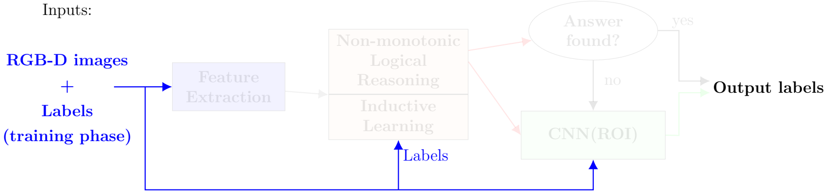
- Deep networks widely used in AI and robotics.
- Limitations of deep network architectures:
  - Large labeled datasets; considerable computational resources; and
  - Representations and mechanisms difficult to interpret.
- Inspiration from human cognition and cognitive systems:
  - Representation, reasoning, and learning tightly coupled.
  - Reasoning with incomplete commonsense knowledge guides interactive and cumulative learning.
  - Principles of relevance and persistence.
- Experimental domains:
  - Estimate object occlusion, and stability of structures.
  - Rearrange objects structures to minimize clutter.
  - Answer explanatory questions (VQA) with limited data.

Architecture Components



Exploit complementary strengths of non-monotonic logical reasoning, deep learning, and decision tree induction.

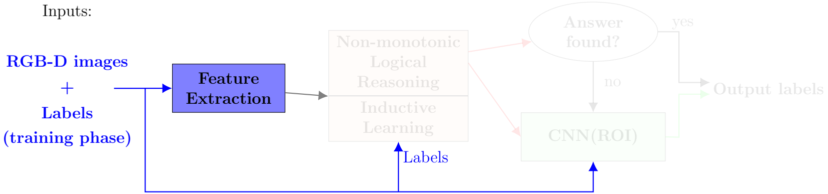
Architecture Components: Input



- Images: images of objects, scenes.
- Labels: object occlusion, stability of structures, answers.



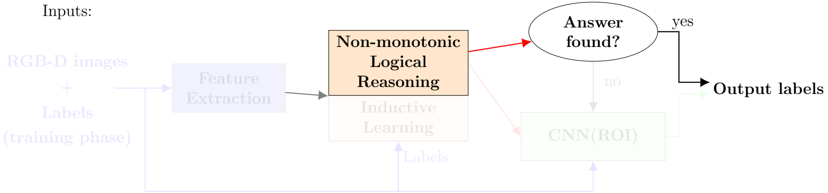

Architecture Components: Feature Extraction



Geometric features extracted from simulated images:

- Spatial relations between objects (above, behind, right of ...).
- Shape and size of objects in the scene.

Architecture Components: Non-monotonic Logic

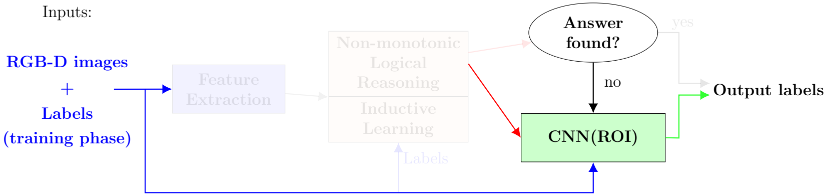


- Input: Extracted features and existing knowledge (including rules learned over time).
- Commonsense reasoning with incomplete knowledge.
- ASP: declarative language; non-monotonic logical reasoning.

$$\neg stable(A) \leftarrow small\_base(A),\ not\ stable(A)$$

- Decision about input image if possible.

Architecture Components: CNN



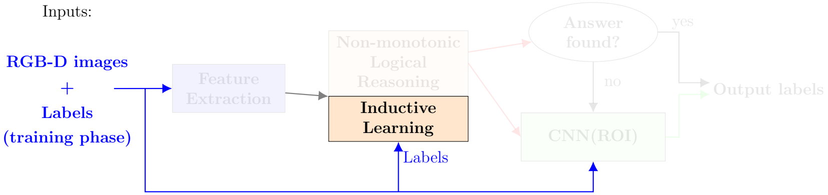
- Attention: ROI selection based on state constraints.

$$stable(A) \leftarrow \neg obj\_rel(above, A, B)$$

$$\neg stable(A) \leftarrow obj\_rel(above, A, B),\ obj\_surface(B, irregular)$$

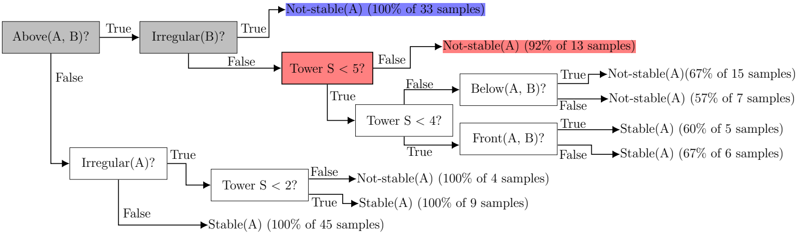
- CNN: Convolutional Neural Network (Lenet and Alexnet).

Architecture Components: Inductive Learning



- Input: Geometric features and figure labels;
- Decision Tree: induction of unknown rules (state constraints);
- Output: Learned rules.

Architecture Components: Inductive Learning



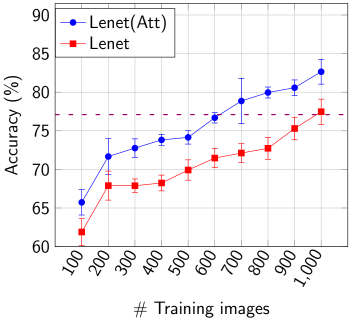
$$\neg stable(A) \leftarrow obj\_rel(above, A, B),\ obj\_surface(B, irregular)$$

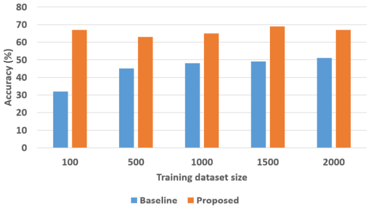
Default knowledge:

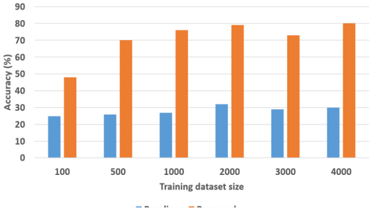
$$\neg stable(A) \leftarrow obj\_rel(above, A, B),\ tower\_height(A, N),\ N \geqslant 5$$

Experimental Results: Scene understanding

- Accuracy increases and training complexity decreases.

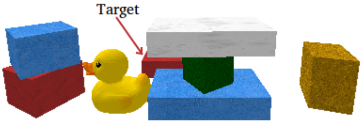






- Generate minimal and correct plans.

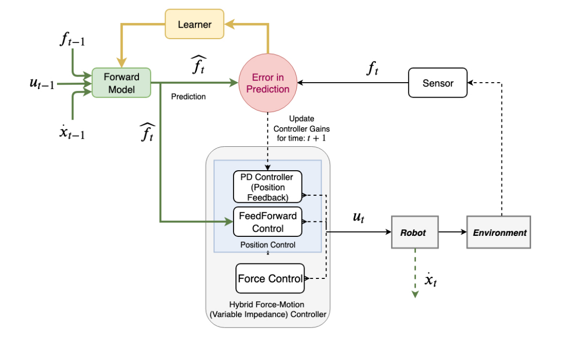
Experimental Results: Decision making



- **Initially**: 64 plans; most incorrect or sub-optimal.
- **Including learned axioms**: 3 correct plans.

- **Without learned axioms:** four times as many plans; six times as much time per plan execution.

Different Problem: Dexterous Manipulation?



- **Status quo**: large datasets, analytic models, joint space control.
- Task-space control; abstract joint trajectories.
- Forward models learned online; variable impedance control.
- Hybrid force-motion controller; compliance.

Conclusions + Future Work

- **Conclusions:**
  - Represent, reason, and learn jointly with different descriptions and mechanisms.
  - Step-wise refinement and separation of concerns simplifies design, increases confidence, promotes scalability.
  - Non-monotonic logical reasoning with commonsense knowledge for reliable and efficient deep learning.
  - Learned state constraints improve decision-making accuracy.

- **Future Work:**
  - Provide intuitive explanations of deep learning models.
  - Explore the interplay between reasoning and learning with different abstractions and reasoning methods.

More Information

- VQA, interactive learning to visually ground spatial relations: **IJCAI-18**, **HAI-18**, **RSS-19** (Best Paper Award Finalist).
- Refinement-based architecture: **NMR-14**, **TRO-15**, **IJCAIwrksp-16**, **AAAISymp-17**, **JAIR-19**.
- Declarative programming and RL for domain dynamics: **ICSR-14**, **ICAPS-17**, **ACS-18**.
- Non-monotonic logic, POMDPs: **ICAPS-08** (Distinguished Paper), **AIJ-10**, **ICDL-12** (Paper of Excellence), **TRO-13**.
- Variable-impedance control: **RSSWrkshp-19** (Best Poster), **Humanoids-19**.

That's all folks!