

LeaPR: Neuro-Symbolic Action Anticipation with Learned Probabilistic Rules

Muyang Yan¹, Maxwell Jacobson¹, Simon Stepputtis², Katia Sycara², Yexiang Xue¹

¹Department of Computer Science, Purdue University, West Lafayette, IN, USA

²The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

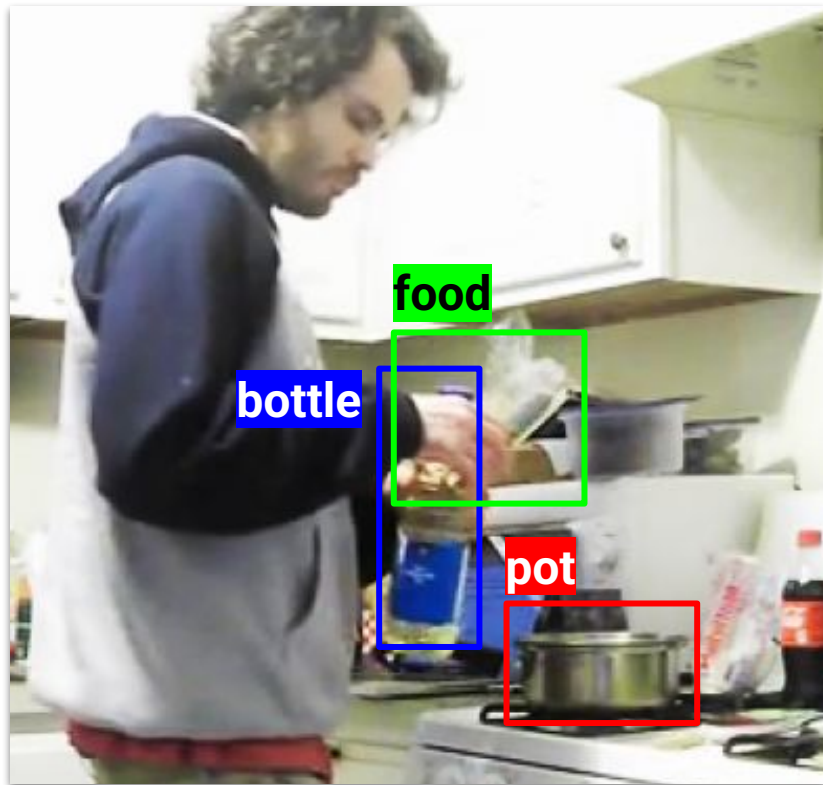
Problem - Action anticipation

Method - LeaPR

Experiments

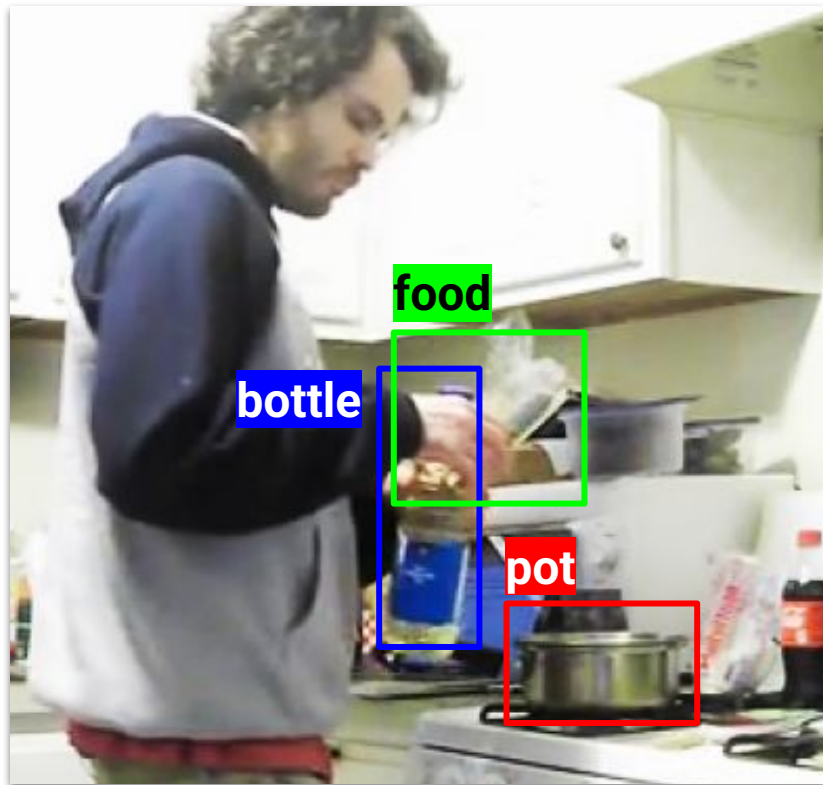
Conclusion

Action anticipation



- A. Eat food
- B. Cook something
- C. Put down bottle

Action anticipation

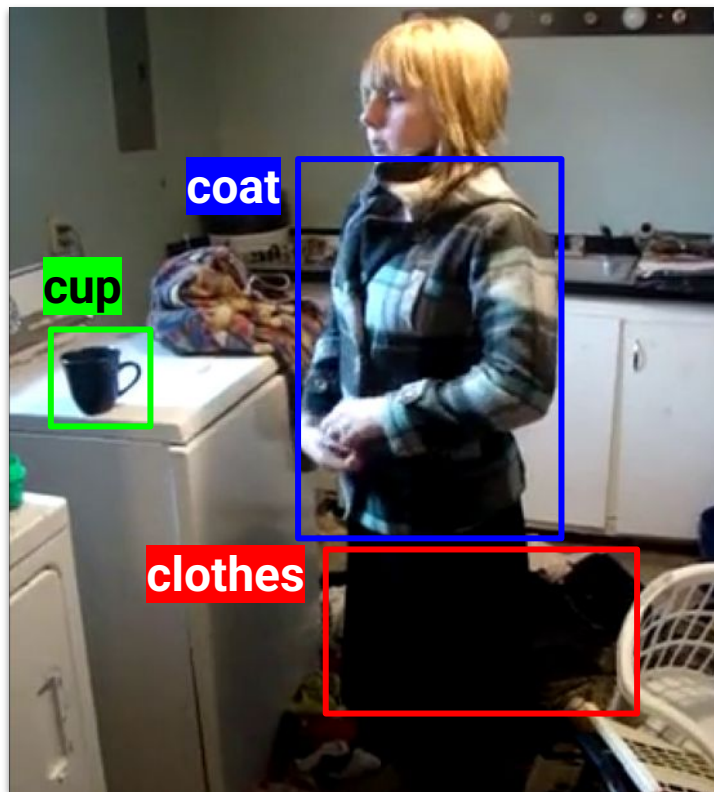


A. Eat food

B. Cook something

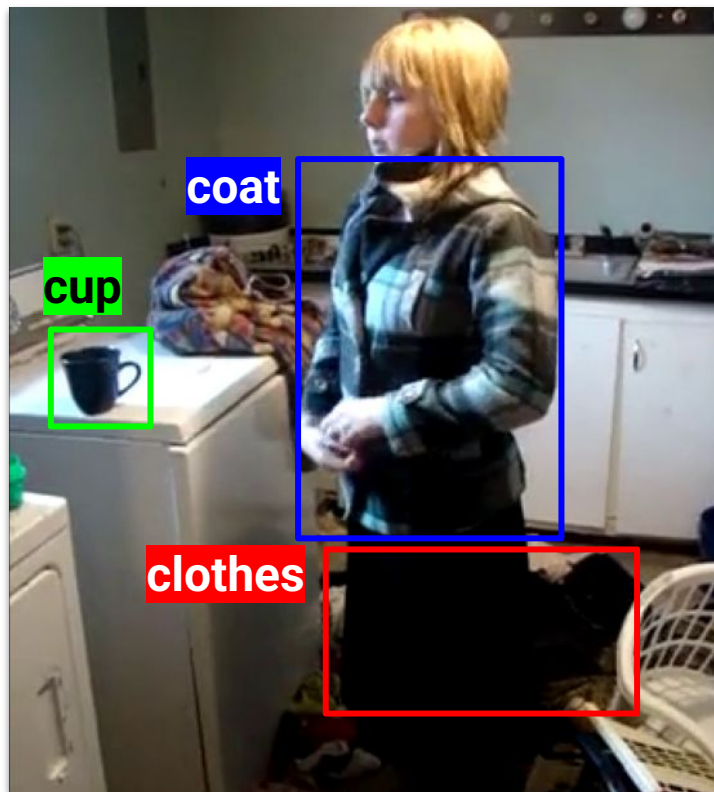
C. Put down bottle

Action anticipation



- A. Pick up cup
- B. Take off coat
- C. Tidy up clothes

Action anticipation



A. Pick up cup

B. Take off coat

C. Tidy up clothes

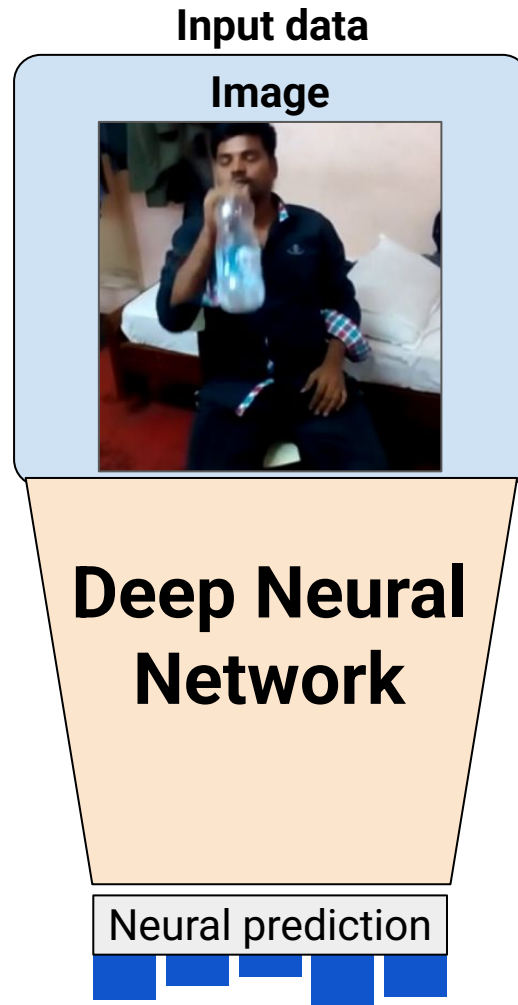
Problem - Action anticipation

Method - LeaPR

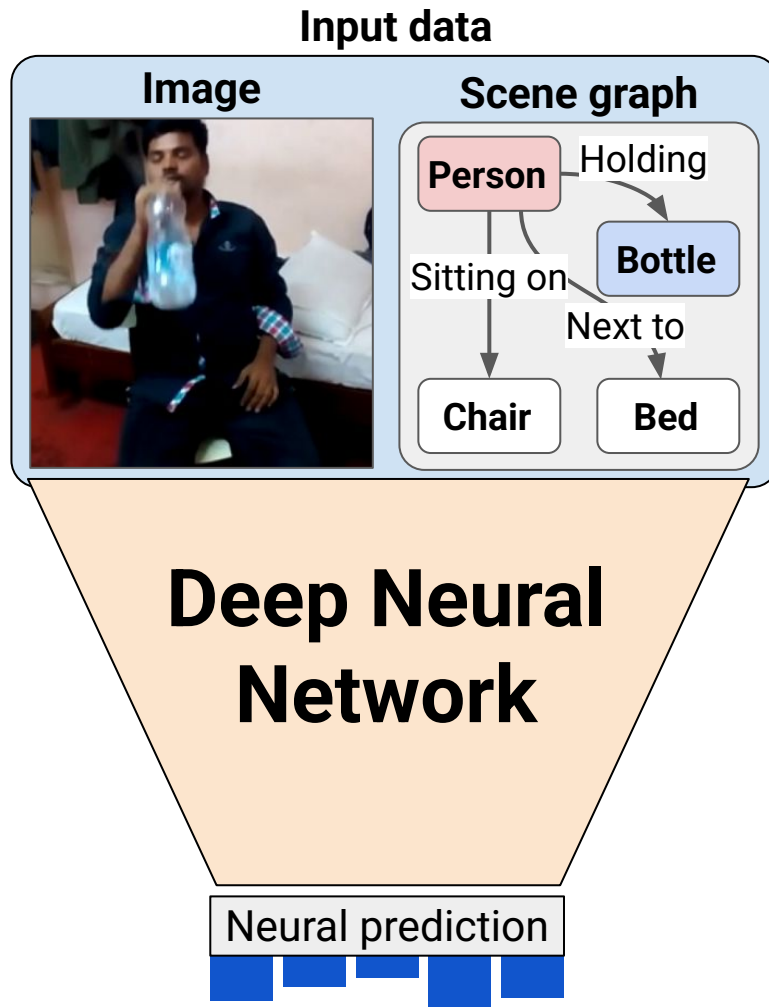
Experiments

Conclusion

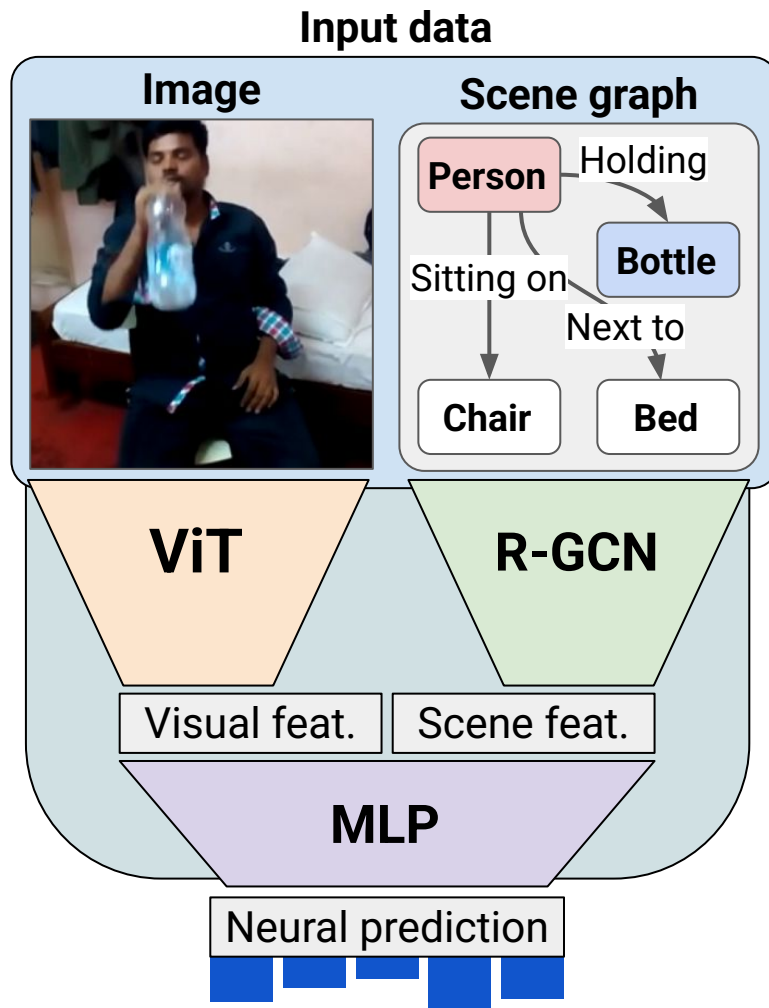
Existing methods

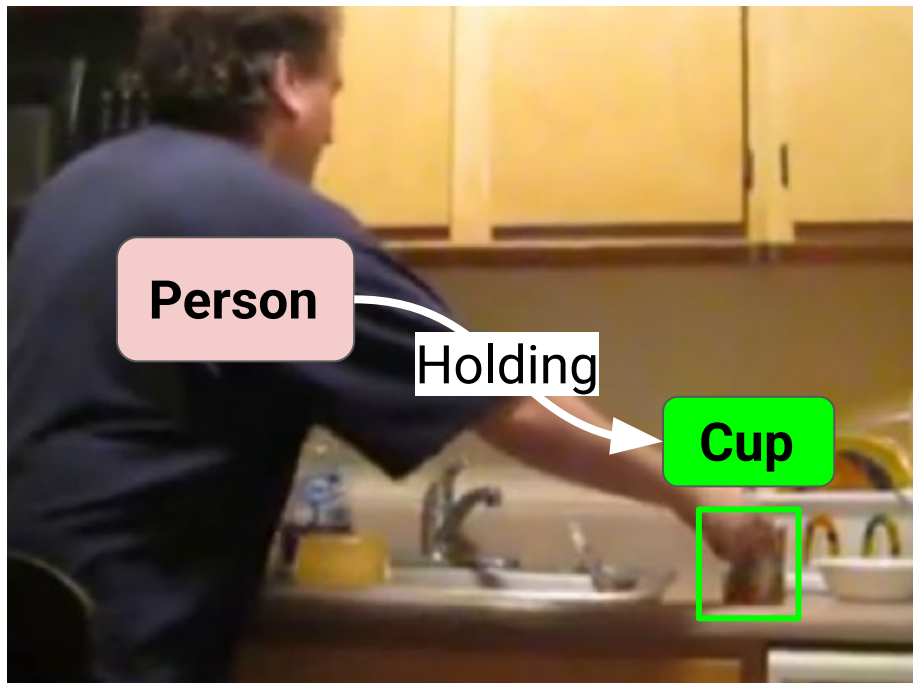


Existing methods



Existing methods

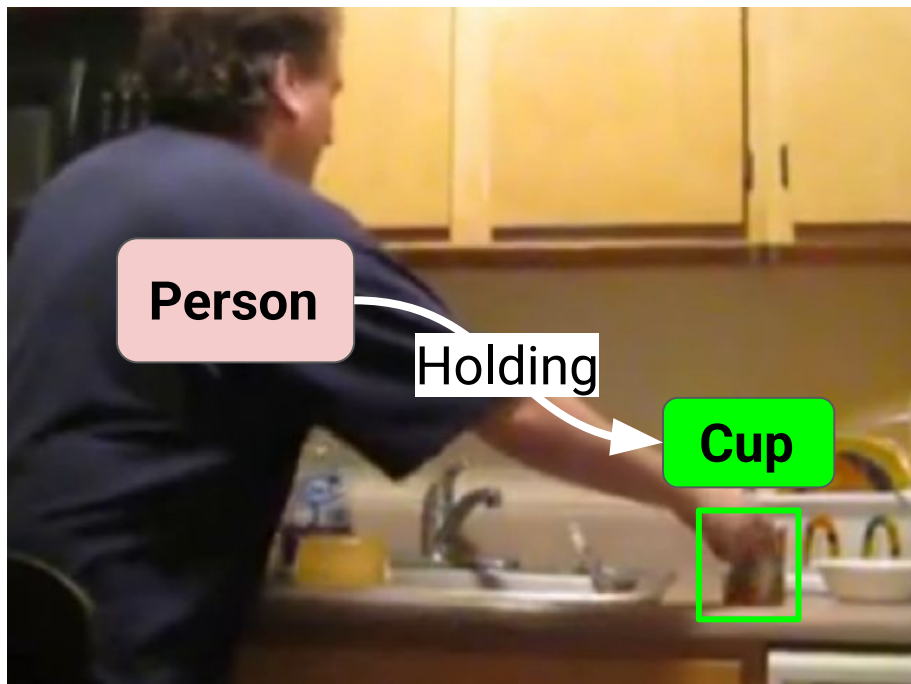




Correct action: ?

Neural network top 5:

1. Take
2. Open
3. Close
4. Hold
5. Wash



Correct action: **Drink**

Neural network top 5:

1. Take ❌
2. Open ❌
3. Close ❌
4. Hold ❌
5. Wash ❌

Problem

Action anticipation is inherently uncertain.

- Hard to predict *exact* correct action.

However, deep learning approaches often predict *obviously infeasible* actions.

- Effect becomes more pronounced as the number of classes increases.

How to reduce this uncertainty?

Rules learned from data

One rule per action in our vocabulary:

“**[Action]** requires some **[Precondition]**”

To **photograph**, the agent must be *holding* a camera

Rules learned from Data

In order to **eat**:



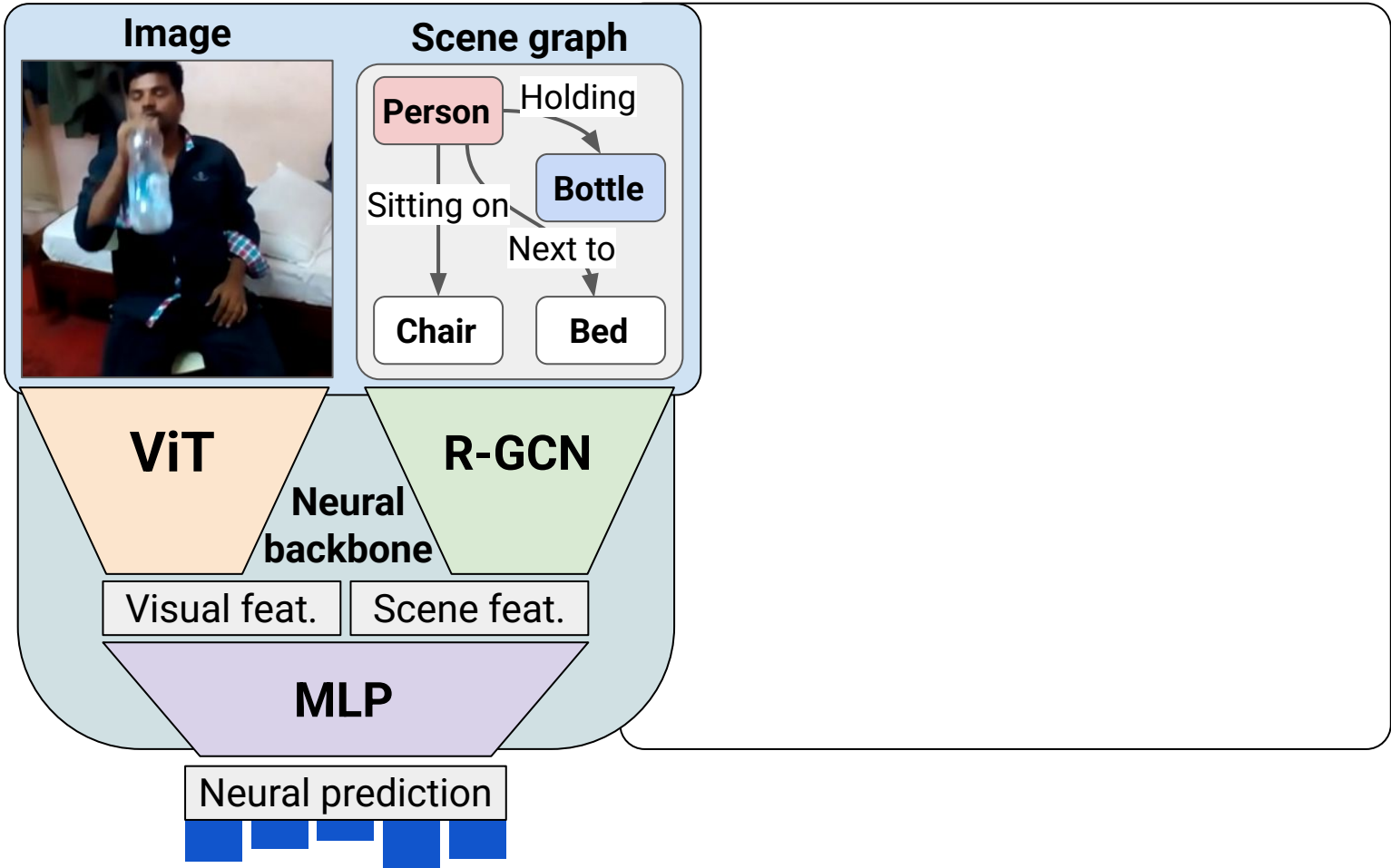
Medicine *in front of* agent

OR



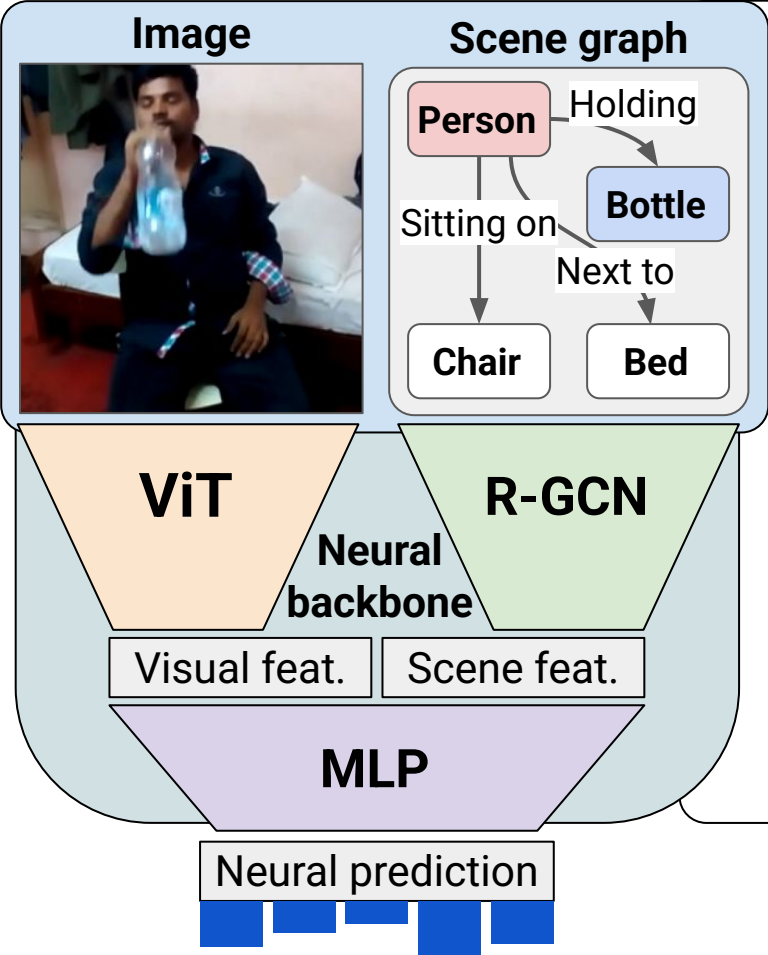
Agent *holding* food

Input data



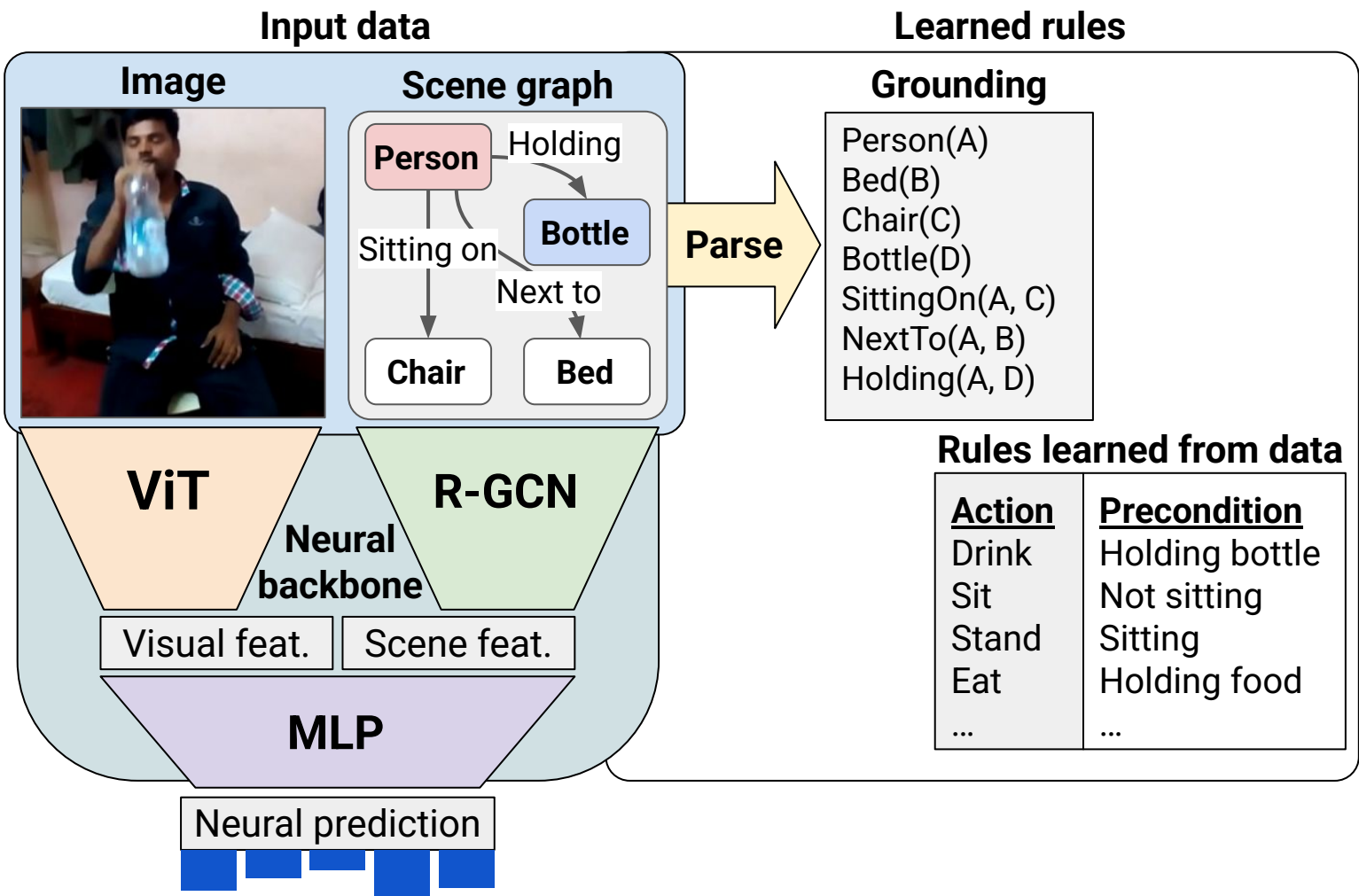
Input data

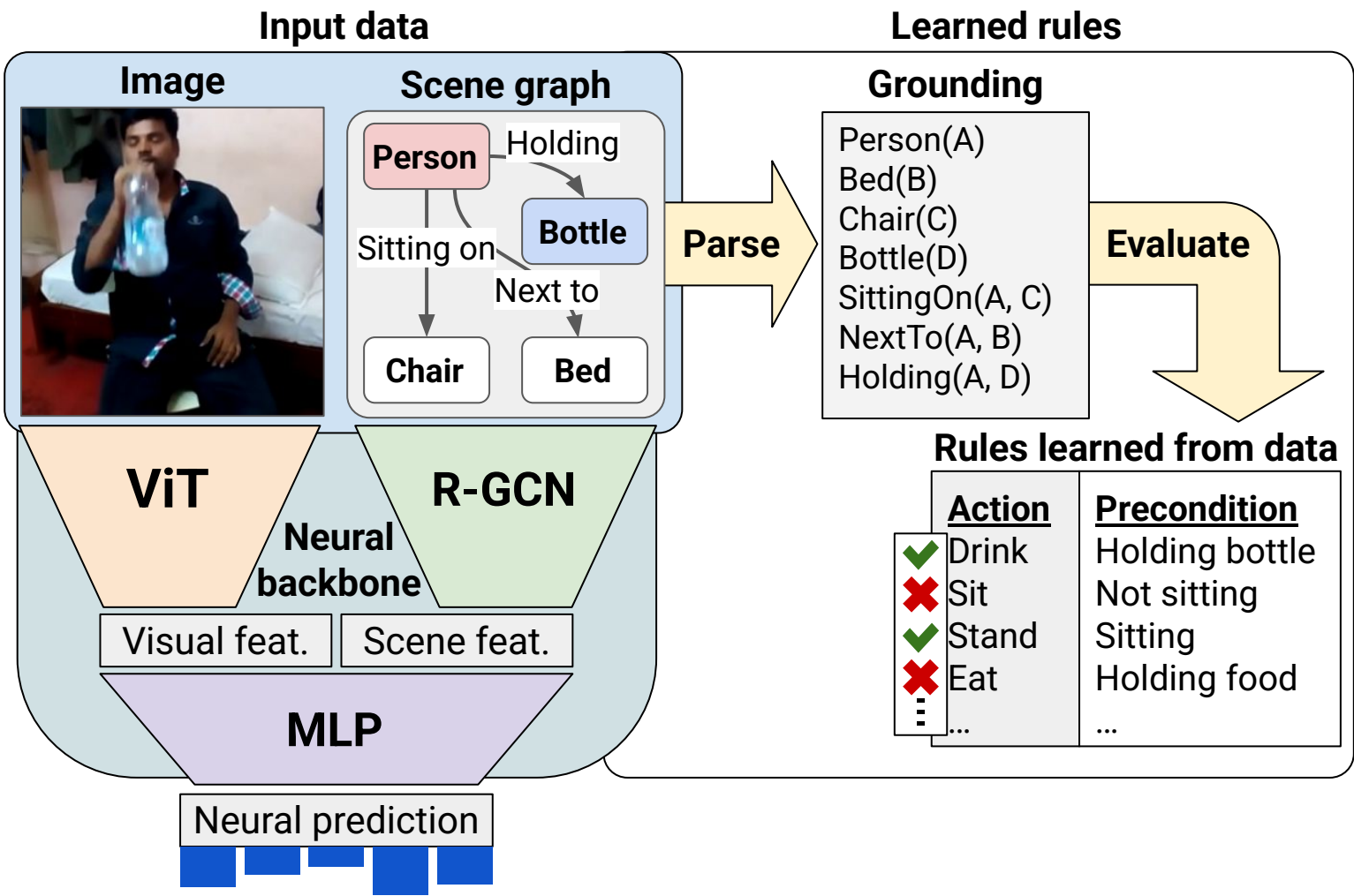
Learned rules

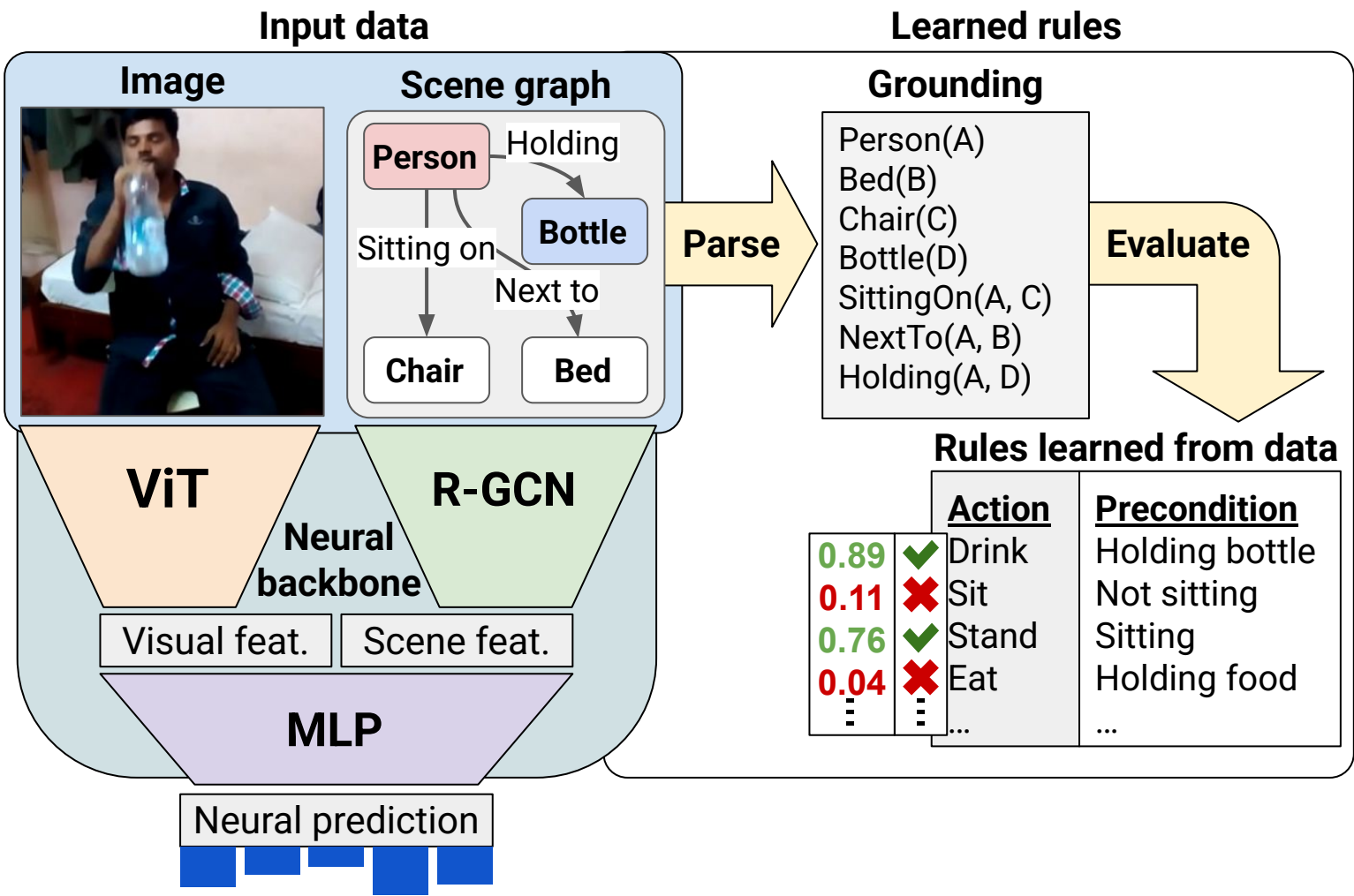


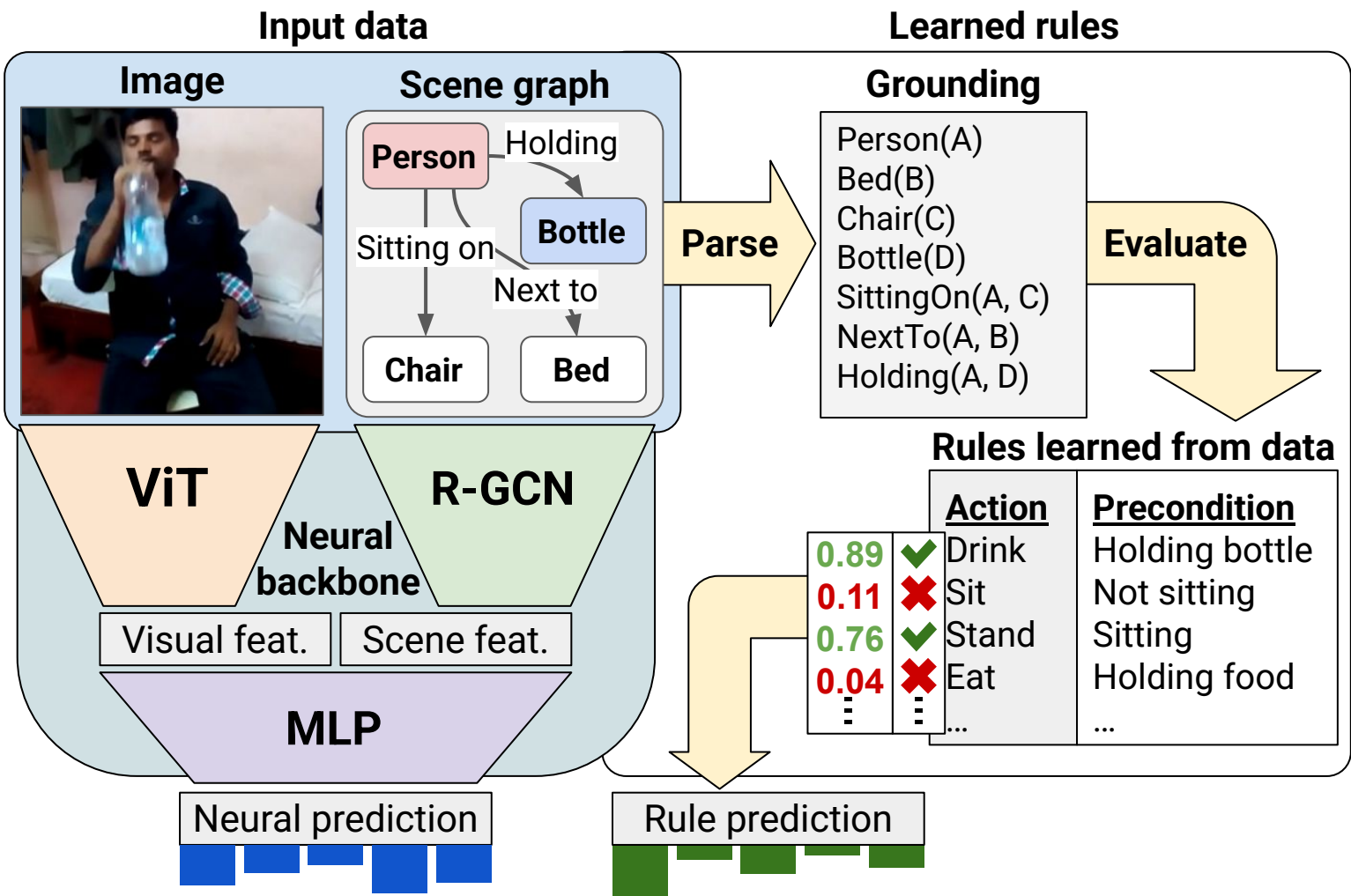
Rules learned from data

<u>Action</u>	<u>Precondition</u>
Drink	Holding bottle
Sit	Not sitting
Stand	Sitting
Eat	Holding food
...	...



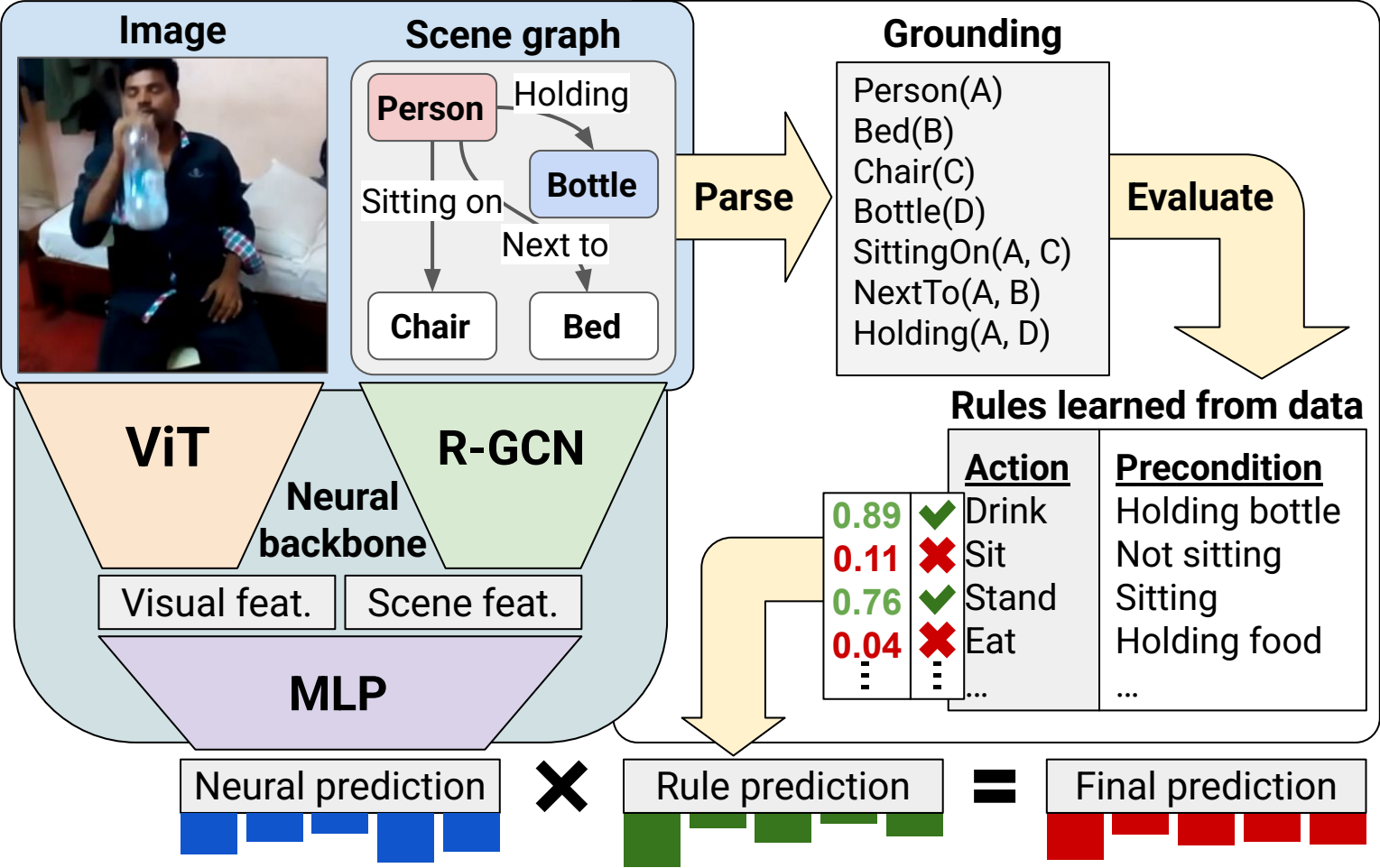


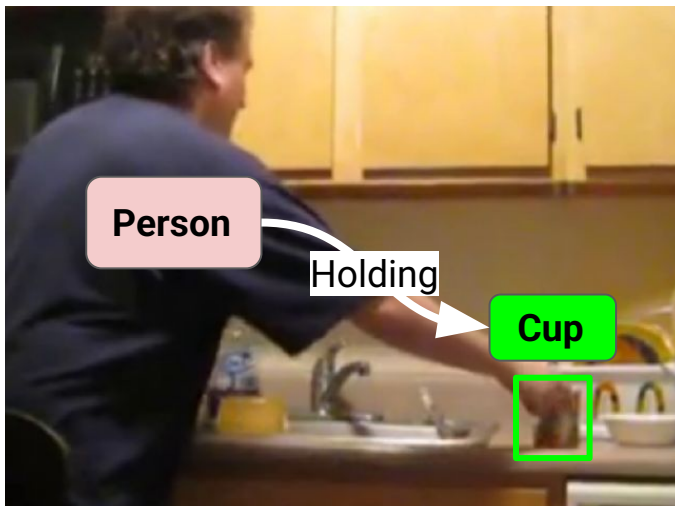




Input data

Learned rules



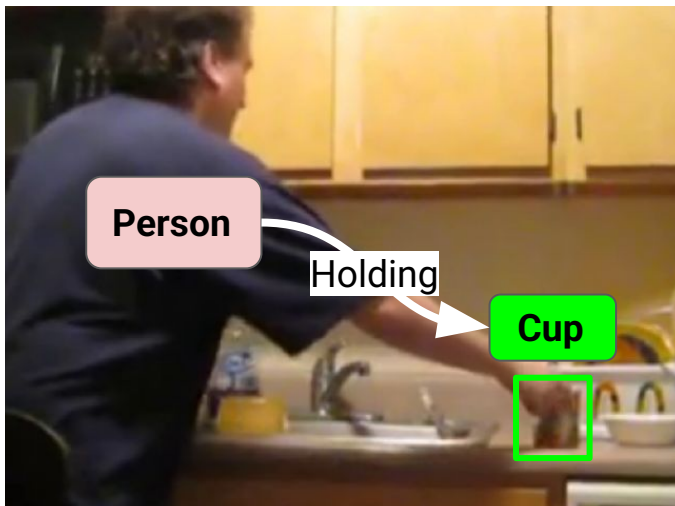


Correct action: **Drink**

Before rules:

1. Take ✗
2. Open ✗
3. Close ✗
4. Hold ✗
5. Wash ✗

After rules:



Correct action: **Drink**

Before rules:

1. Take ❌
2. Open ❌
3. Close ❌
4. Hold ❌
5. Wash ❌

After rules:

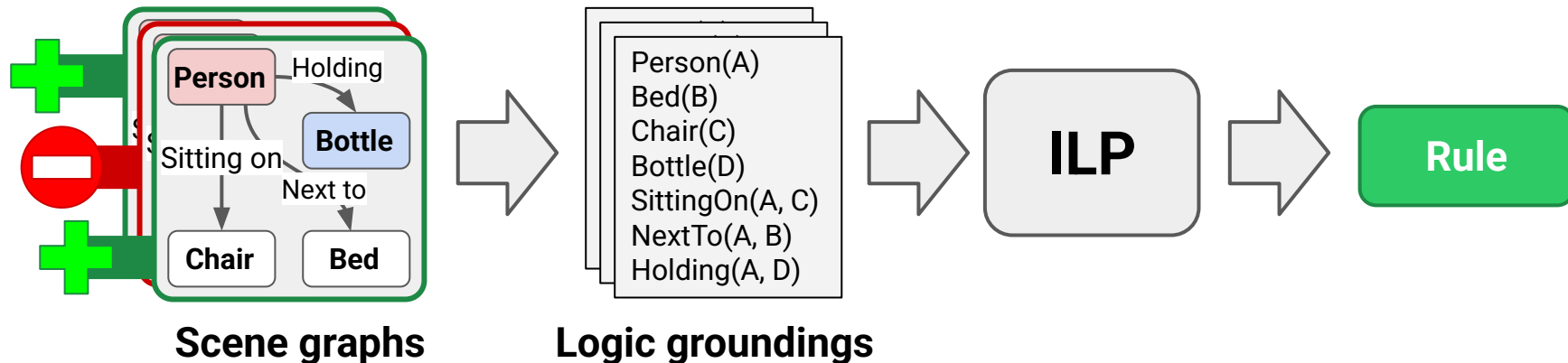
1. **Drink** ✓
2. Put
3. Pour
4. Open
5. Sit

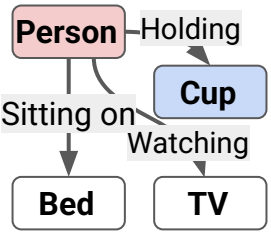
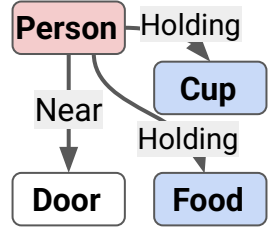
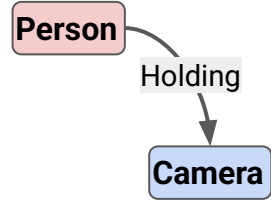
How are the rules learned?

Inductive Logic Programming (ILP)

For each action:

- Input: Dataset of examples, labeled **Positive(+)** or **Negative(-)**
- Output: A **rule** entailing as many positives and as few negatives as possible



Example		Label	
		Drink	
		Drink	
		Take Photo	

According to the data, what is required to **Drink**?

Example		Label	
	<pre> graph TD Person[Person] -- Holding --> Cup[Cup] Person -- Sitting on --> Bed[Bed] Person -- Watching --> TV[TV] </pre>	Drink	+
	<pre> graph TD Person[Person] -- Holding --> Cup[Cup] Person -- Near --> Door[Door] Person -- Holding --> Food[Food] </pre>	Drink	+
	<pre> graph TD Person[Person] -- Holding --> Camera[Camera] </pre>	Take Photo	-

According to the data, what is required to **Drink**?

To Drink:

-> Person holding a Cup

Problem - Action anticipation

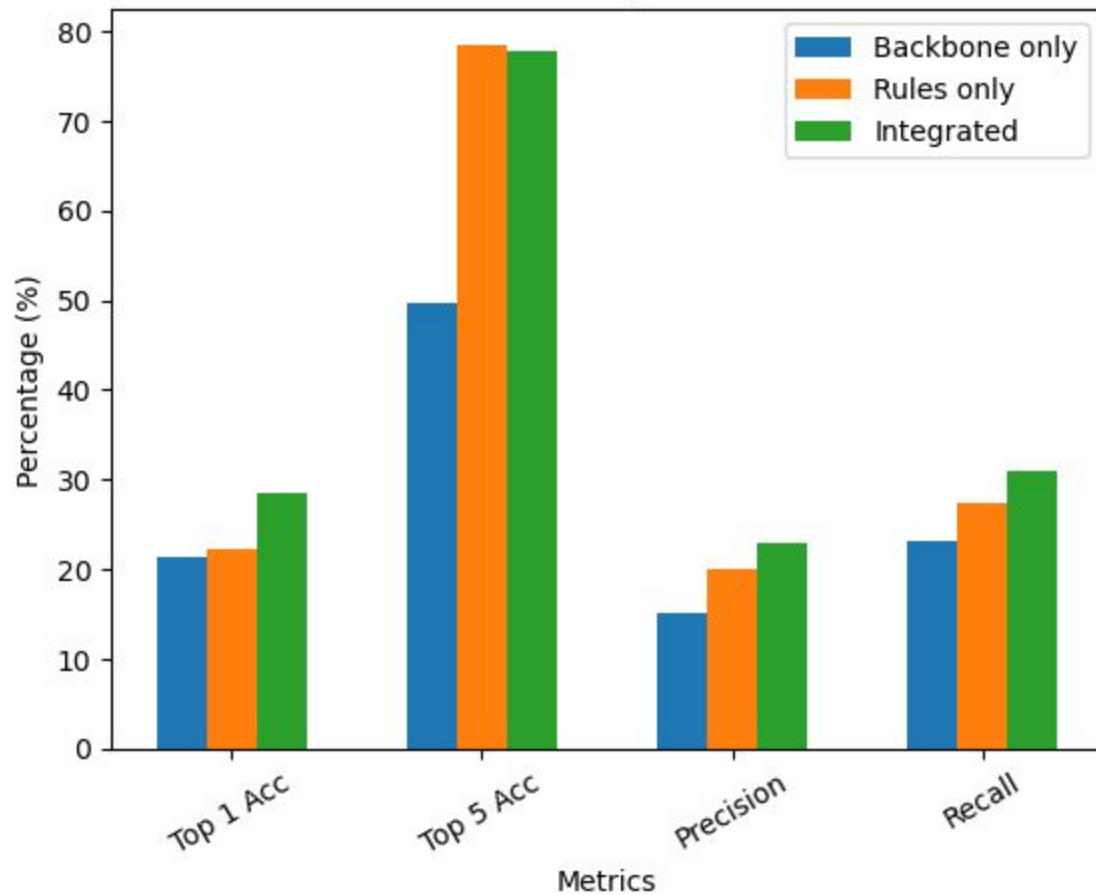
Method - LeaPR

Experiments

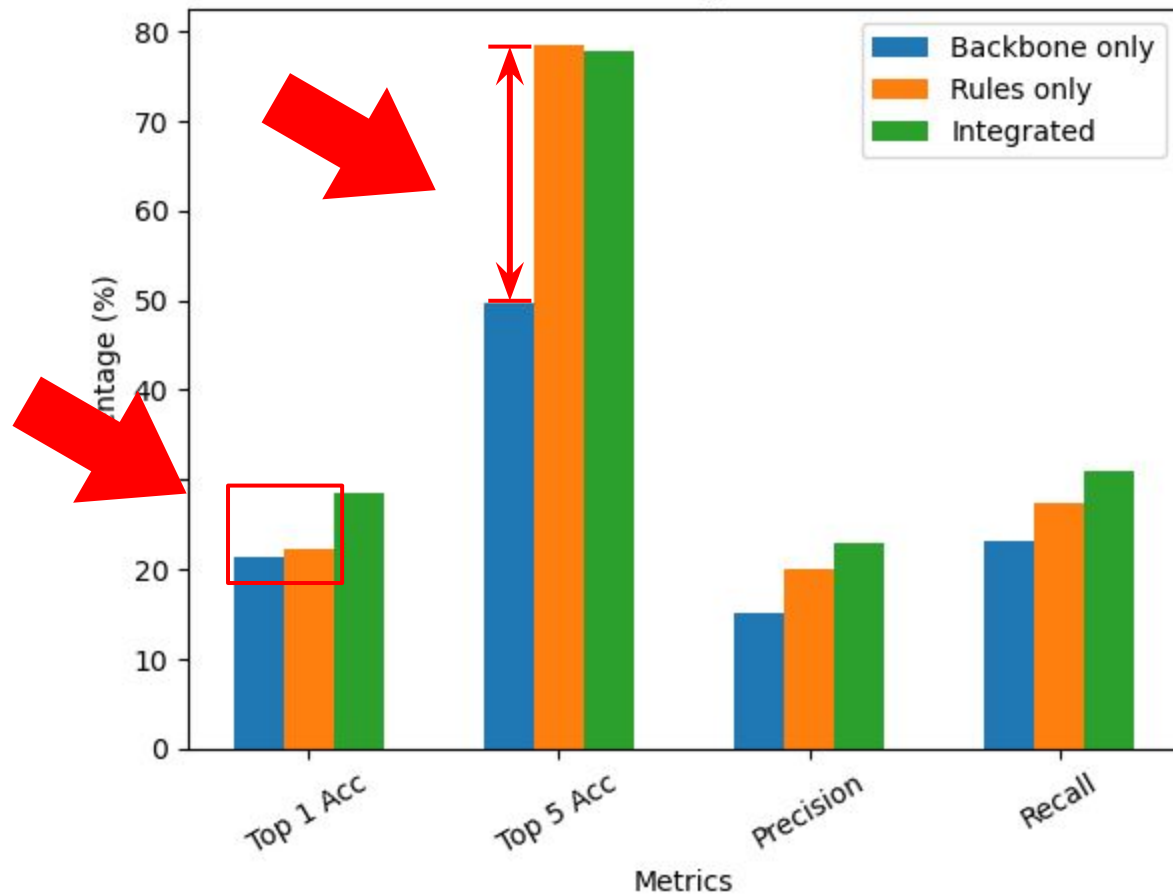
Conclusion

Architecture	Top 1 Acc	Top 5 Acc	Precision	Recall
Neural backbone only, no rules	21.4	49.6	15.0	23.2
Just rules, no neural backbone	22.2	78.5	19.9	27.4
Integrated	28.5	<i>77.7</i>	22.9	31.0

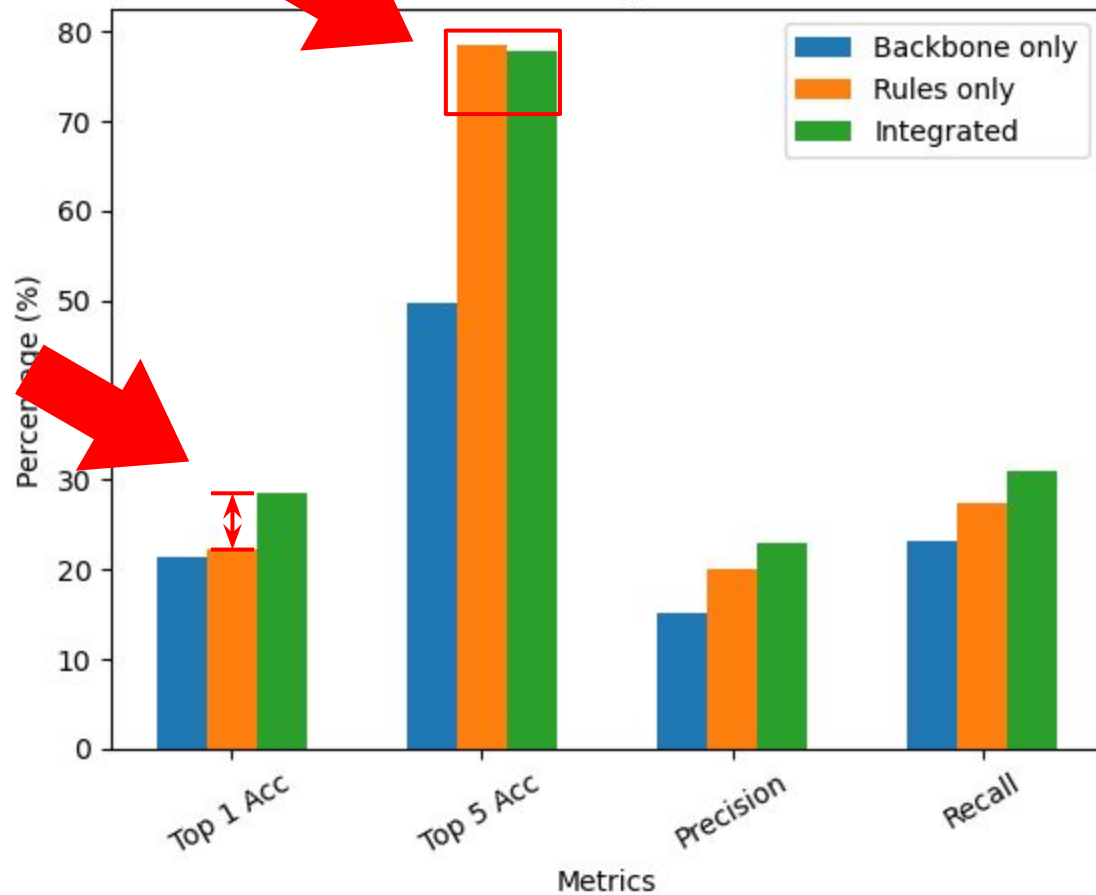
Ablation study of LeaPR



Ablation study of LeaPR



Ablation study of LeaPR



Problem - Action anticipation

Method - LeaPR

Experiments

Conclusion

Conclusion

Action anticipation is a very uncertain problem.

Reduce uncertainty by combining ML and ILP.

Rules support the prediction by highlighting the **feasible** candidates.

Neural network captures unstructured nuances to **refine** predictions.

Conclusion

ML and ILP both try to learn a predictive model from data.

- In ML, it's a set of weights for a neural network
- In ILP, it's a set of rules.

Integrated, they do better than alone.

What other problems may benefit from this integration?

Acknowledgements



Prof. Yexiang Xue



Prof. Katia Sycara



Max Jacobson



Dr. Simon Stepputtis

Contact me:
Muyang Yan
yan427@purdue.edu

Extra slides

Probabilistic Rules

Consider the rule:

Talk requires agent to be *holding* a cellphone

Precision: 75%

In 75% of instances of agent holding cellphone, Talk was next action

Recall: 100%

In 100% of the instances of talking in the dataset, agent was holding a cellphone