Robot Learning Ensuring Stability and Robustness to Irreversible Events

Pietro Falco
pietro.falco@unipd.it

ELLIIT Symposium on Robot Learning

Lund, 20/11/2025

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DEGLI STUDI DI PADOVA

A Robot in Every Home

- Well known article by Bill Gates in 2007
- *“Robots will become as pervasive as Personal Computers.”*
- After 18 year, what is the status?

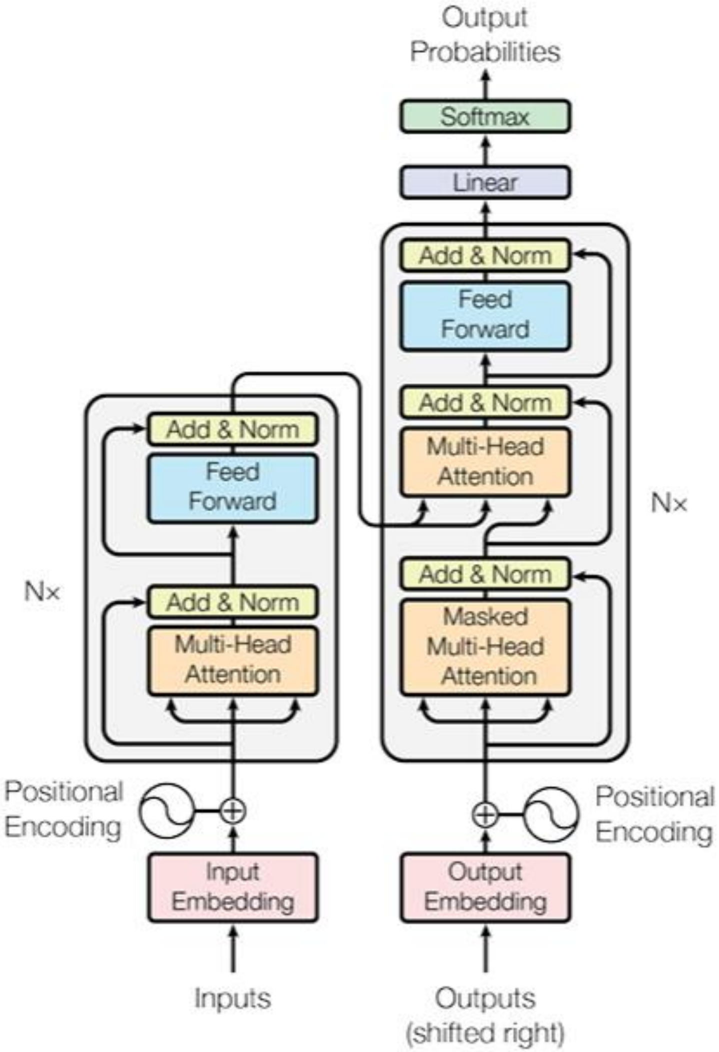
Gates, Bill. "A robot in every home." *Scientific American* 296.1 (2007): 58-65.

Modern Robotics

Transition from production lines to **unstructured, anthropic** environments

We want that nontechnical users can be able to instruct the robot

Transformer-based Architectures



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

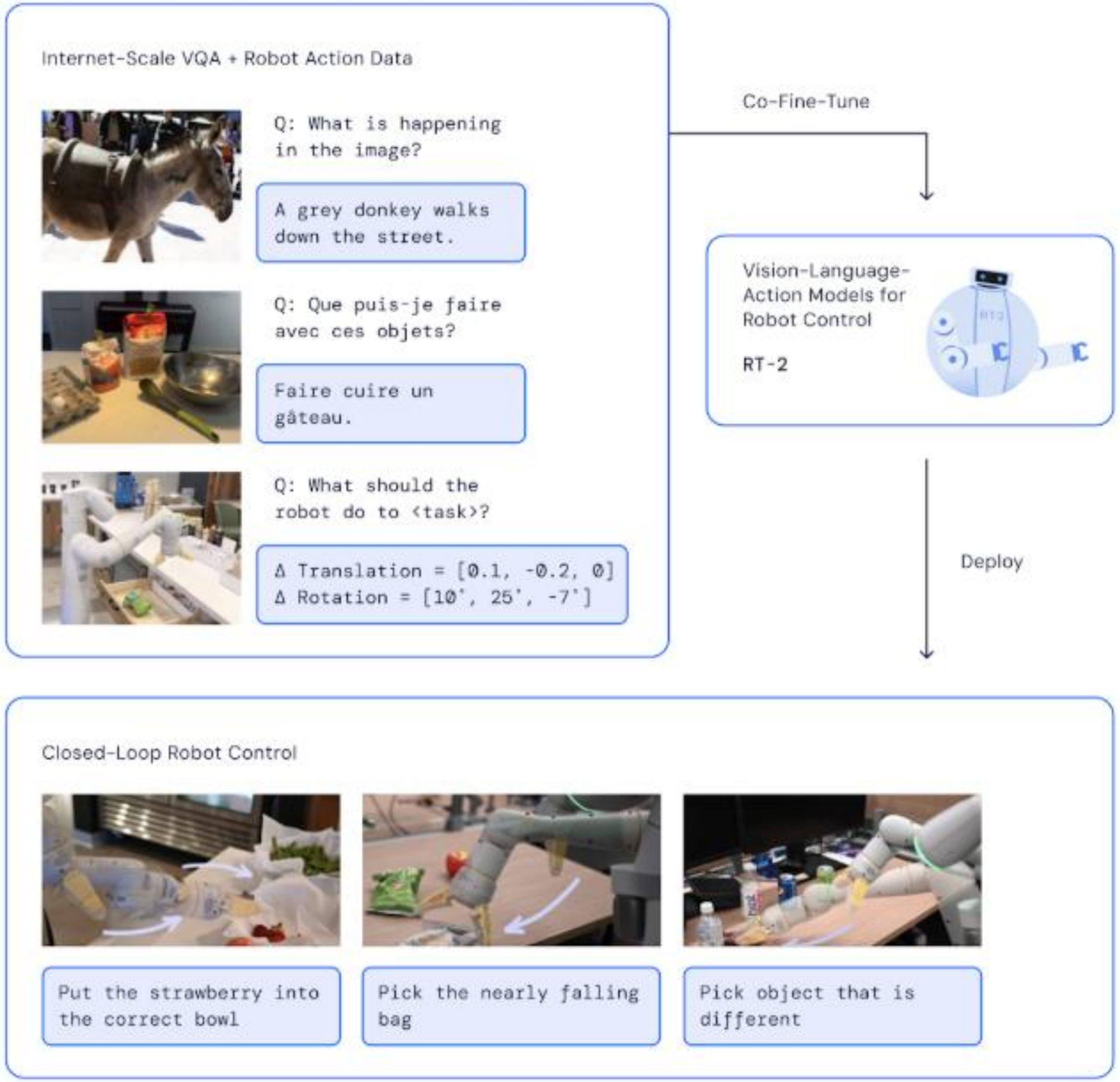
Focus on relevant parts of the input sequence

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Example of Technologies

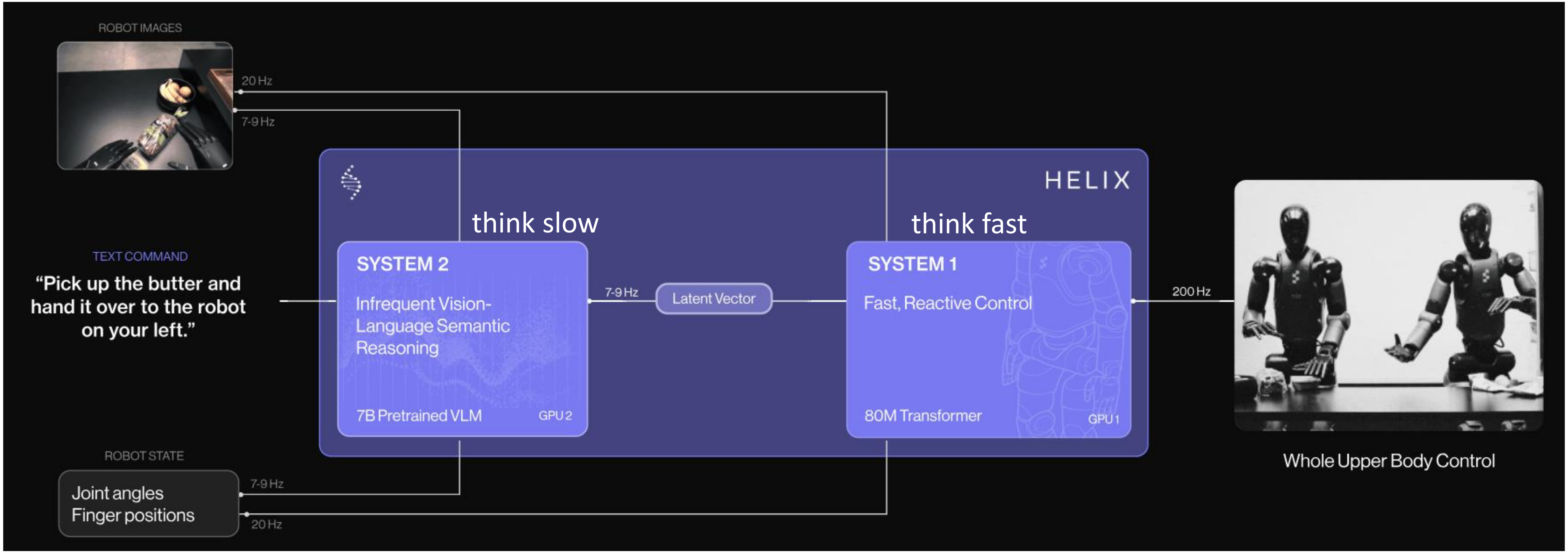
Model Type	Modalities	Input/Output	Training Method	Example Models
LLM (Large Language Model)	Language	Text → Text	Self-supervised learning	GPT, BERT, LLaMA
VLM (Vision-Language Model)	Vision + Language	Image + Text → Text	Self-supervised learning	CLIP, Flamingo
VLAM (Vision-Language-Action Model)	Vision + Language + Action	Image + Text → Action	Supervised imitation learning	RT-2, HELIX, PI0, OpenVLA

VLAM - Robot Transformer 2 (RT-2)

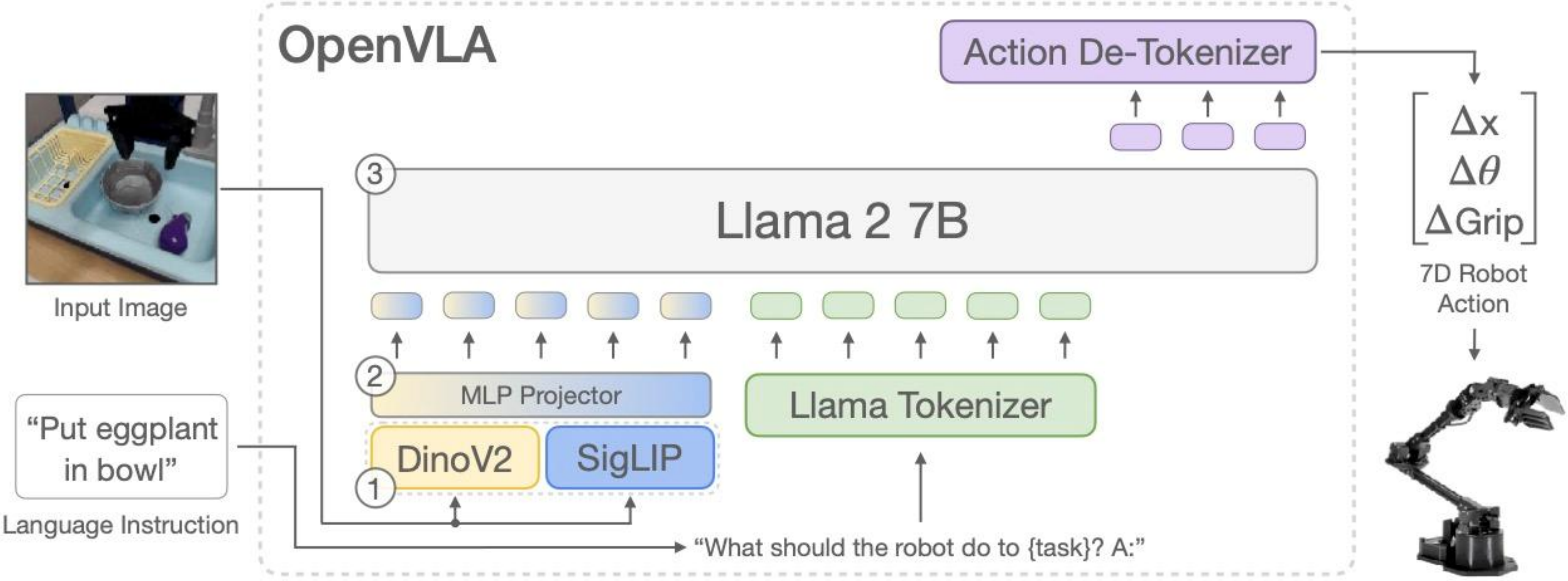


Output are high-level actions, separate motion planner

Visual Language Action Models (VLAM) - Helix



A Open Source Architecture



Example of First Industrial Applications

Figure AI Robot – Pilot @BWM

Transformer-based arcitectures:

- Strong breakthrough in human-robot interface and General AI
- Still work to do on safe adaptation in real environments

Reinforcement Learning can give a contribution on adaptation skills

RL in Manipulation - Sim2real

Akkaya, Ilge, et al. "Solving rubik's cube with a robot hand." *arXiv preprint arXiv:1910.07113* (2019).

RL in Locomotion

Boston Dynamics and AI & Robotics Institute , 2025

It uses **human motion capture** to learn natural motion pattern

RL is used for adaptation:

- To learn terrain-aware gait adjustments
- To correct for uncertainties or dynamics not modeled in the MPC
- Sim2real transfer
- **To fine-tunes** behaviors to match reference under robot dynamics

Some Key Challenges in Real-World Adaptation

1. High number of rollouts, potentially also in sim2real

   Many works in the scientific community to reduce rollouts on the real robot (model-based reinforcement learning, sim2real)

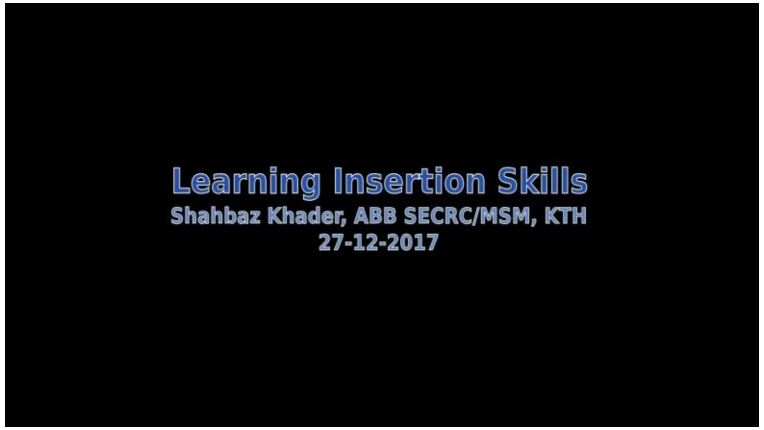
2. Irreversible events

3. Ensuring formal guarantees during the different rollouts, e.g. stability certification

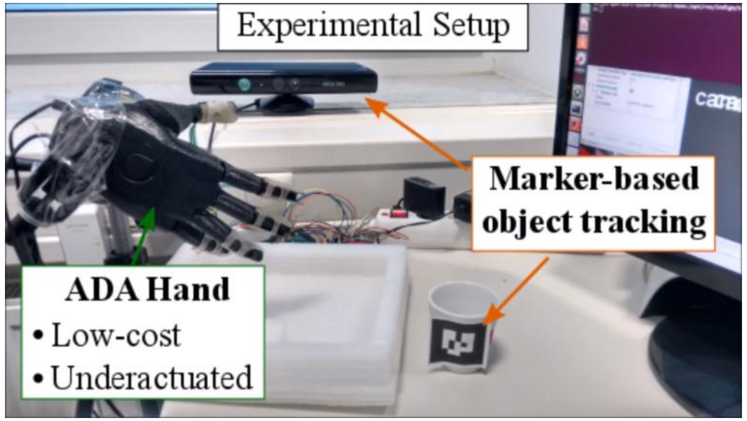
4. Express reward/cost functions without technical skills

Irreversible Events

- Typically, it is assumed that the robot can try an **infinite number of rollouts, returning to the initial state after each rollout and continue the exploration phase**



- Is it always possible to **keep exploring in reality**?



In real applications irreversible events happen, which can make impossible for the robot to keep learning autonomously.

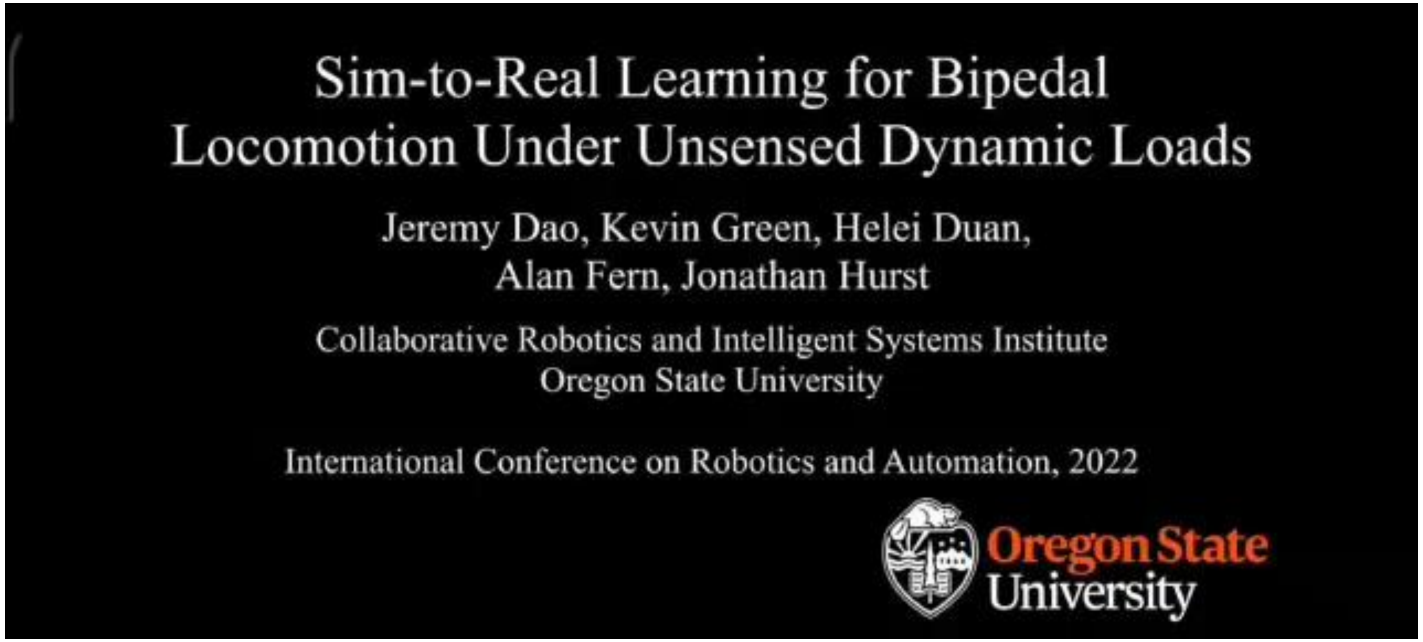
Falco, Pietro, Abdallah Attawia, Matteo Saveriano, and Dongheui Lee. "On policy learning robust to irreversible events: An application to robotic in-hand manipulation." *IEEE Robotics and Automation Letters* 3, no. 3 (2018): 1482-1489.

Irreversible events

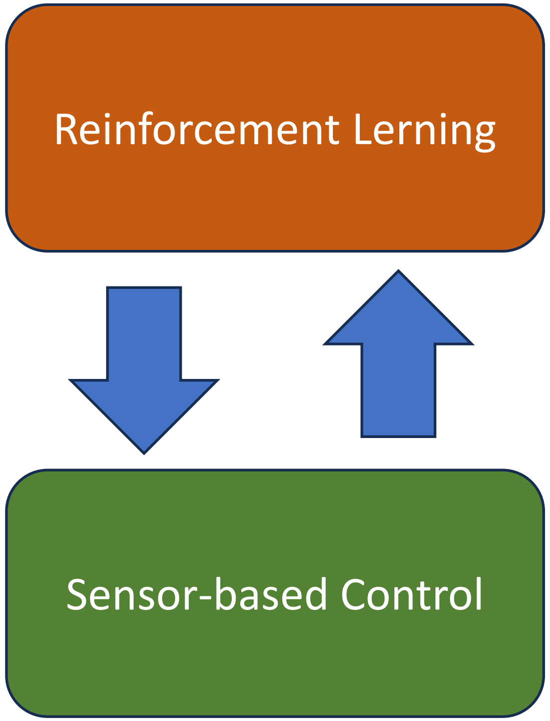
Question: how do we increase the robustness to irreversible events?

Sim2Real based solution

- The topic of irreversible events is not well-covered in the robotics literature
- Similar approaches are sim-to real
- Introducing disturbances in simulation, with the aim to get a more robust policy
- Typical in Locomotion
- In real world unpredictable scenario it is difficult to totally ovoid fine tuning

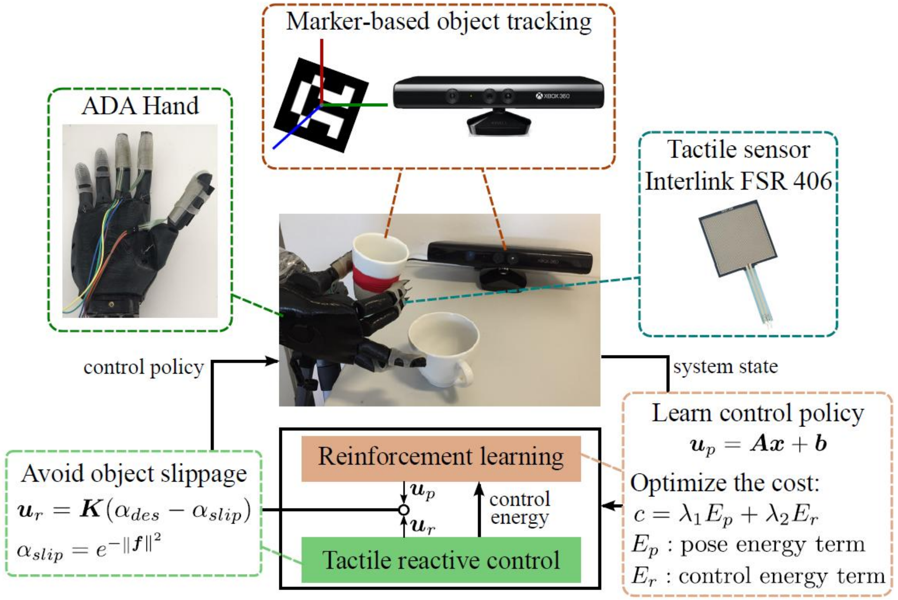


How do we tackle the problem?



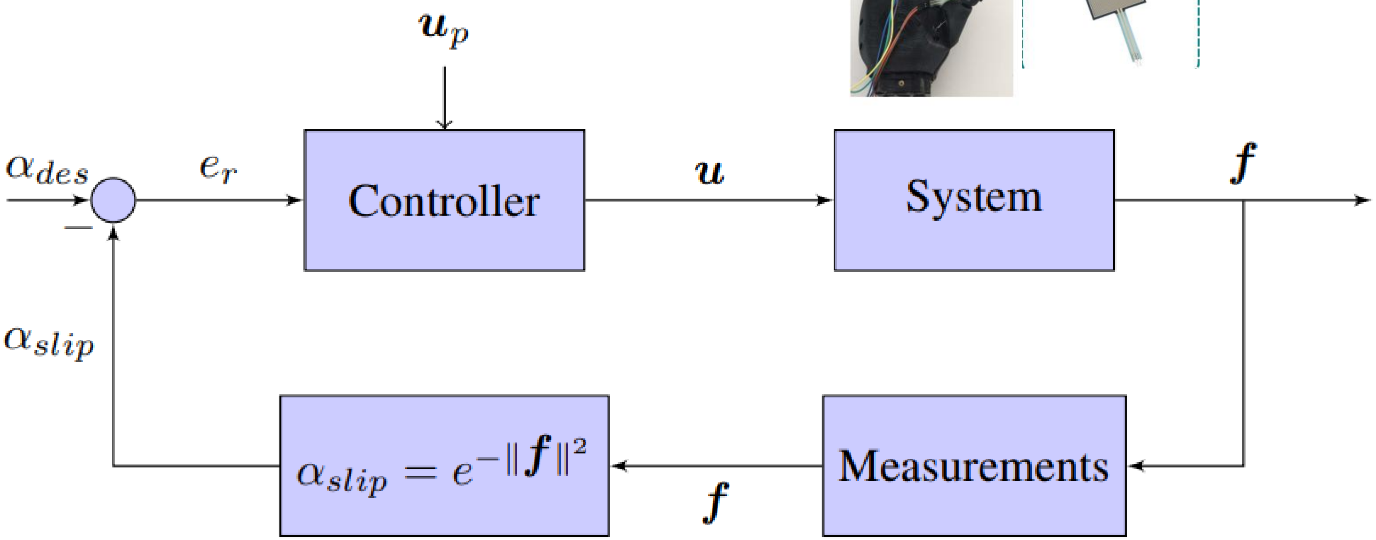
The RL **learns the task and, at the same time, minimize the need of intervention of reactive control** in future executions.

Intervenes to avoid irreversible events when needed



Falco, Pietro, Abdallah Attawia, Matteo Saveriano, and Dongheui Lee. "On policy learning robust to irreversible events: An application to robotic in-hand manipulation." *IEEE Robotics and Automation Letters* 3, no. 3 (2018): 1482-1489.

Reactive Control



$$
\begin{aligned}
\boldsymbol{u} &= \boldsymbol{u}_p + \boldsymbol{u}_r, \\
\boldsymbol{u}_r &= \boldsymbol{K} e_r, \\
e_r &= \alpha_{slip} - \alpha_{des},
\end{aligned}
$$

$$E_r = |\alpha_{slip} - \alpha_{des}|$$

u: vector of motor commands
u_p: command from RL layer
u_r: command from reactive control
α_{slip}: slipping factor
E_r: reactive pseudoenergy

Reinforcement Learning

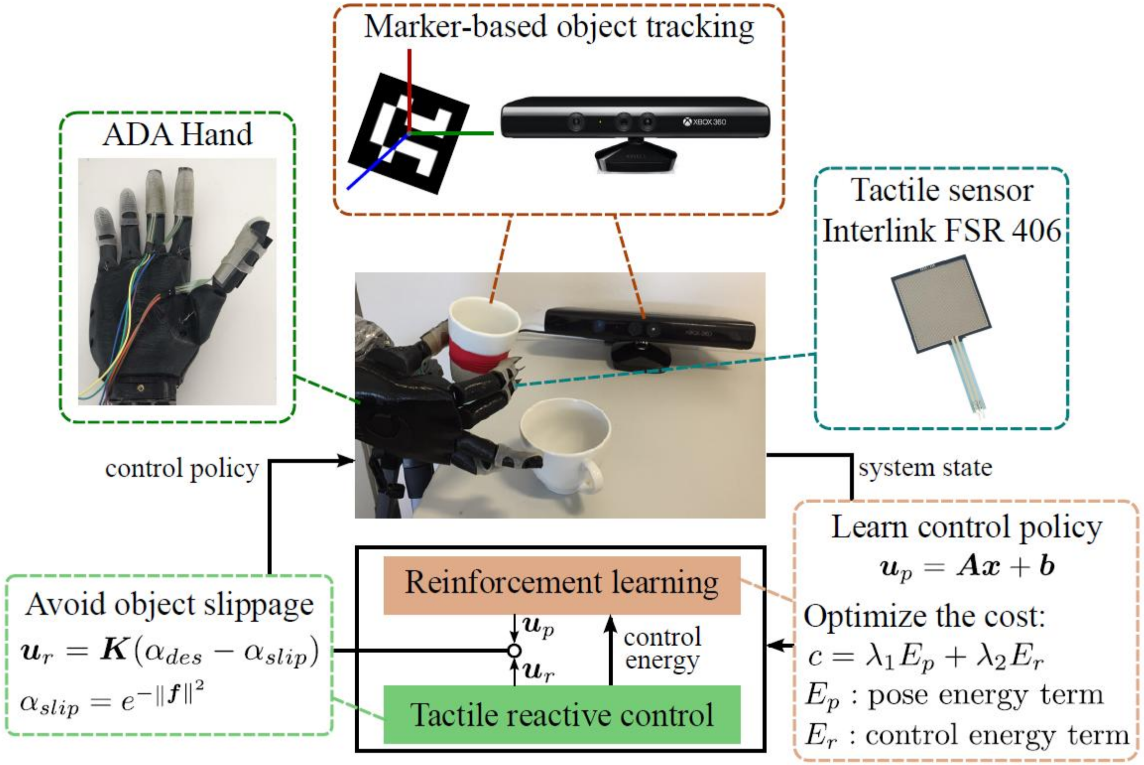
$$c = \lambda_1 E_p + \lambda_2 E_r,$$
$$E_p = 1 - e^{-||\phi - \phi_{des}||^2},$$
$$E_r = |\alpha_{slip} - \alpha_{des}|,$$

E_p is related to object orientation error

E_r is related to reflexes intervention

The RL algorithms minimizes both!

The system learn to avoid the need of reflexes

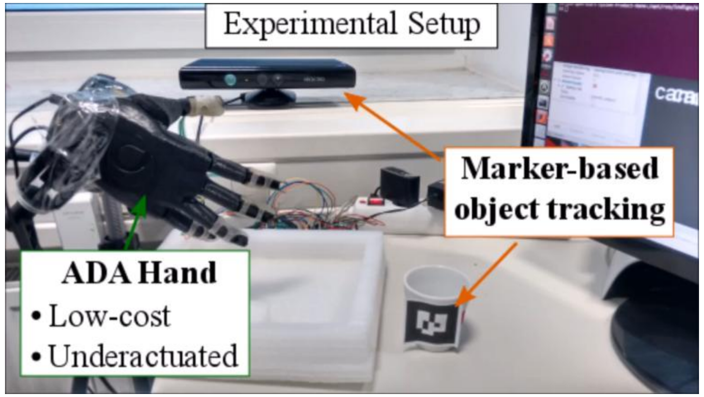


Falco, Pietro, Abdallah Attawia, Matteo Saveriano, and Dongheui Lee. "On policy learning robust to irreversible events: An application to robotic in-hand manipulation." *IEEE Robotics and Automation Letters* 3, no. 3 (2018): 1482-1489.

Experimental Results – PILCO RL algorithm

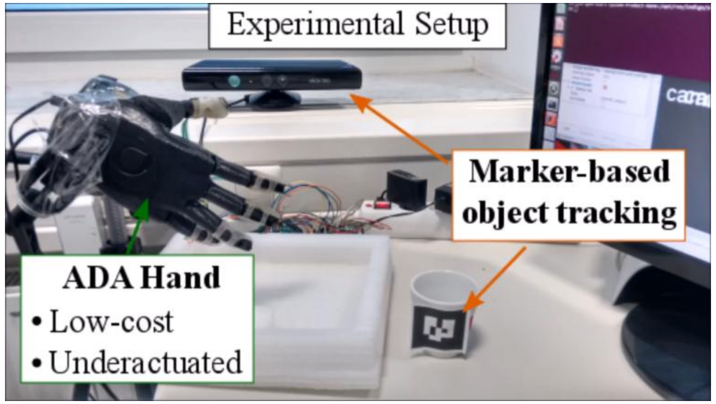
Visual and tactile data. no synergy

Trials [#] \ Rollouts [#]	1	2	3	4	5	6	7	8	9	10	11
1	1.0										
2	0.94	0.18	0.11	0.04	0.07	0.08	0.17	0.19	0.13	0.25	0.19
3	0.92	0.12	0.18	0.07	0.15	0.13					
4	0.87	0.07	0.13	0.05	0.11	0.09	0.15				
5	0.78	0.19	0.13	0.15	0.21	0.15	0.13	0.2	0.17	0.11	0.05
6	0.72	0.23									
7	0.51	0.11	0.09	0.09	0.03						
8	0.51	0.08	0.09	0.18	0.11						
9	0.5	0.04	0.15	0.03							
10	0.44	0.03									



Learning-control synergy

Trials [#] \ Rollouts [#]	1	2	3	4	5	6	7	8	9	10	11
1	0.94	0.17	0.15	0.04	0.07	0.17	0.04	0.02			
2	0.92	0.17	0.04	0.02							
3	0.78	0.07	0.05	0.04	0.02						
4	0.75	0.04	0.04	0.03							
5	0.75	0.17	0.04	0.03							
6	0.69	0.08	0.05	0.02							
7	0.6	0.14	0.06	0.15	0.01						
8	0.56	0.1	0.17	0.16	0.14	0.15	0.04	0.02			
9	0.5	0.12	0.1	0.1	0.05	0.03					
10	0.49	0.2	0.09	0.15	0.16	0.1	0.18	0.17	0.16	0.09	0.19

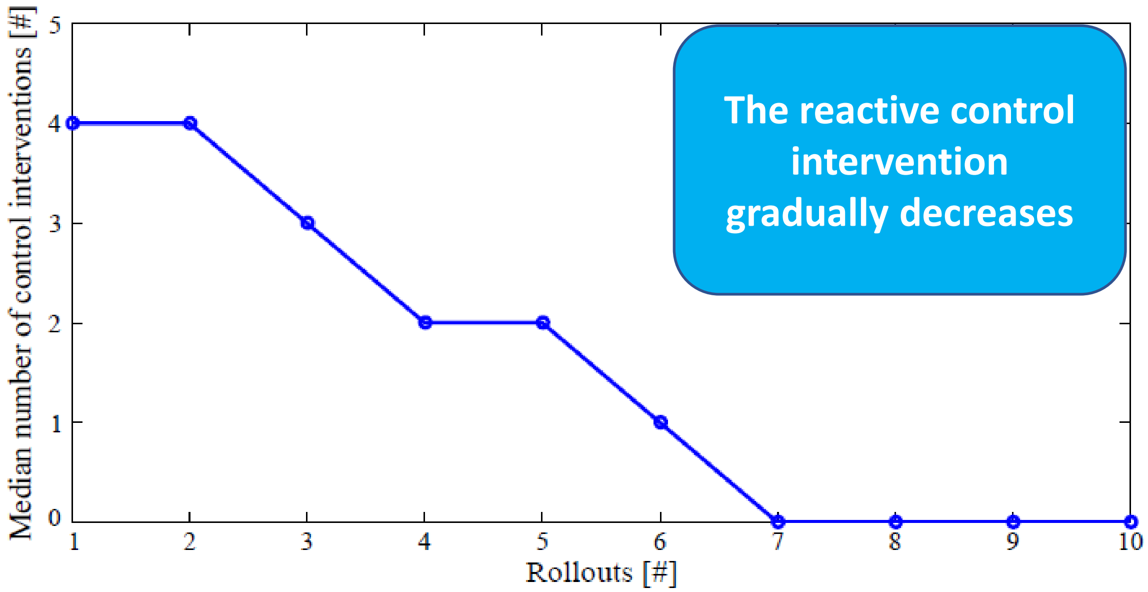


Behavior of the Reactive Control Interventions

Term related to the task execution

Term related to reactive control correction

- The Reinforcement learning module minimizes the cost $c = E_p + E_r$
  - The need for the reactive control correcting the trajectories decreases over the rollouts
  - The stronger is the correction from the reactive control, the larger is E_r



Falco, Pietro, Abdallah Attawia, Matteo Saveriano, and Dongheui Lee. "On policy learning robust to irreversible events: An application to robotic in-hand manipulation." *IEEE Robotics and Automation Letters* 3, no. 3 (2018): 1482-1489.

Conclusion

- Synergy between slow-thinking reactive layer and reinforcement learning strategies can enable the learnability

Some Key Challenges in Adaptation

1. High number of rollouts, potentially also in sim2real

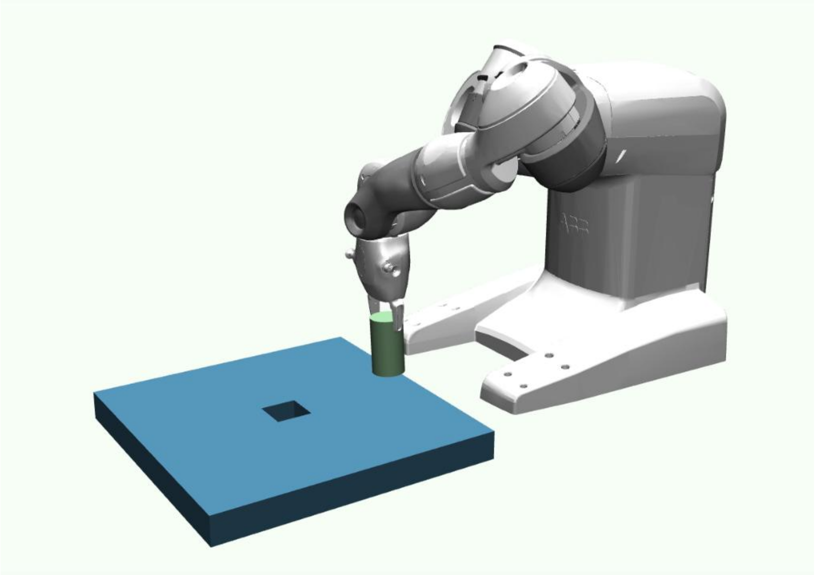
   Many works in the scientific community to reduce rollouts on the real robot (model-based reinforcement learning, sim2real)

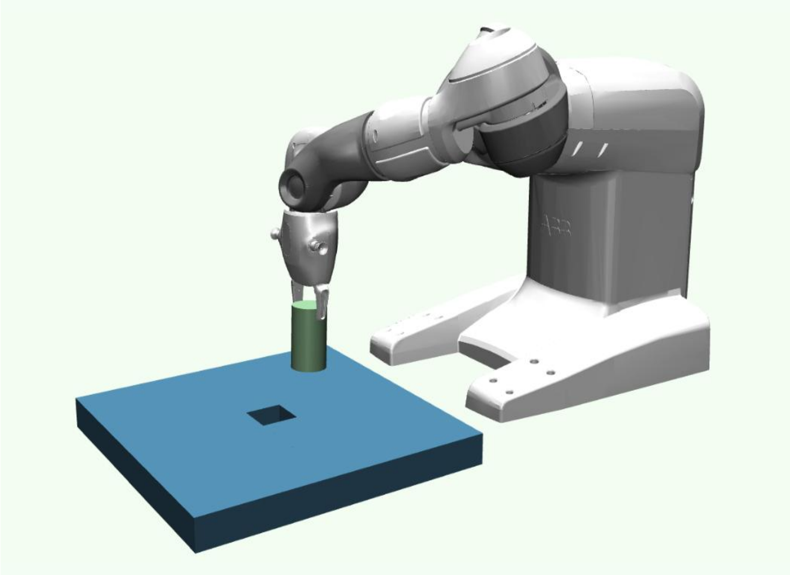
2. Irreversible events

3. **Ensuring formal guarantees (stability) during the different rollouts**

4. Express reward/cost functions without technical skills

Do We Need Stability Certification in RL?





Classical methods Any trial could diverge away from the goal

- No measure of safety or predictability in the initial stages
- Some predictability in the later stages

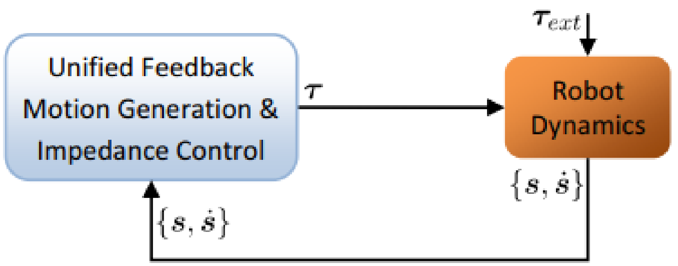
Proposed approach RL <u>with</u> stability guarantee

CEM Every trial tends to the goal with a guarantee

- Stability maintained despite random policy initialization and exploration
- A good measure of safety and predictability in all stages

Stability in Learning Contact-rich Tasks

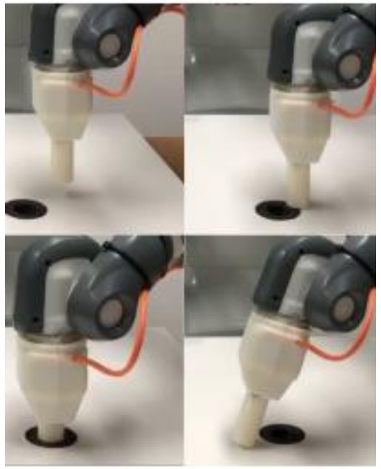
- In an RL context, stability corresponds to a guarantee that any rollout is bounded in state space and tends **to the goal position** demanded by the task



- The policy should ideally have only one stable equilibrium point at the goal

- In **real-world applications,** it is important to make the exploration predictable

- We need to choose a suitable policy representation and policy search



SA Khader, H Yin, **P Falco**, D Kragic , Stability-guaranteed reinforcement learning for contact-rich manipulation - IEEE Robotics and Automation Letters, 2020

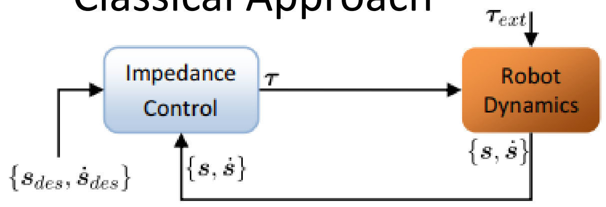
Policy Representation: I-Mogic

- We use I-Mogic a control policy structure for contact rich tasks

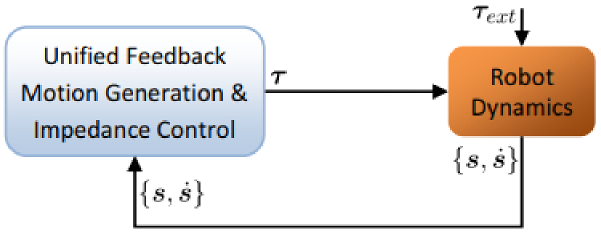
$$\mathbf{u} = -\boldsymbol{S}^0 s - \boldsymbol{D}^0 \dot{s} - \sum_{k=1}^{K} w^k(s)[\boldsymbol{S}^k(s - s^k) + \boldsymbol{D}^k \dot{s}]$$

- S^k are stiffness matrices, D^k damping matrices, $w^k(s, l^k)$ are weighting coefficients
- It is a combination of mass-spring-damper systems

Classical Approach



Used in our work



- The origin of the state space is the only equilibrium point and it is stable if the following conditions are met

$$\boldsymbol{S}^0 = (\boldsymbol{S}^0)^T \succ 0 \quad \boldsymbol{D}^0 \succ 0$$

$$\boldsymbol{S}^k = (\boldsymbol{S}^k)^T \succeq 0 \quad \boldsymbol{D}^k \succeq 0 \quad l^k > 0 \quad \forall k = 1, \dots, K$$

S. M. Khansari-Zadeh, K. Kronander, and A. Billard, "Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control," Proceedings of Robotics: Science and Systems X (RSS 2014), vol. 10, p. 2014, 2014.

Policy Search – Wishart Distribution

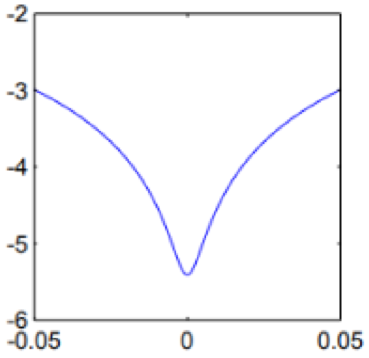
- We have to ensure that the robot **adopts only stable policies at each rollout**
- We use the **Wishart Distribution** to sample only symmetric positive-definite matrices

$$S \sim \mathcal{W}_D(S|W, \nu)$$

$$S \in \mathbb{R}^{D \times D}$$
$$W \in \mathbb{R}^{D \times D}$$

We used the following reward:

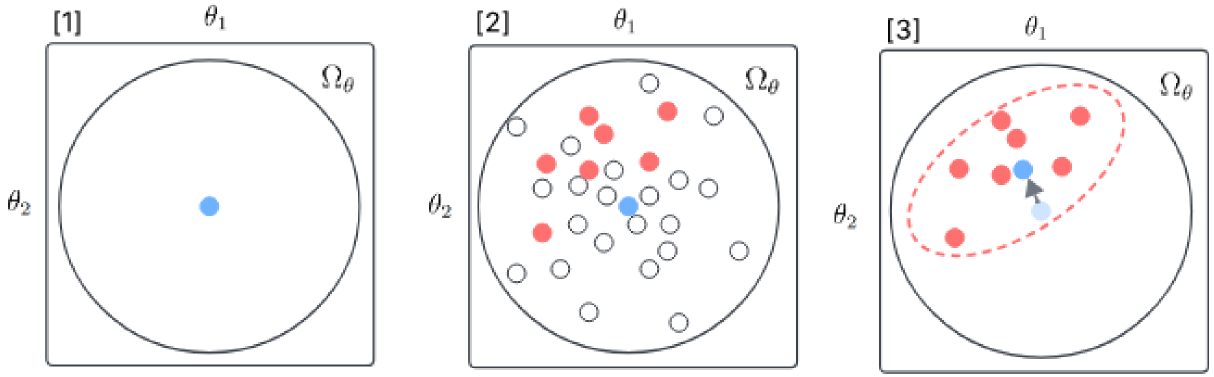
$$r_\ell(d) = wd^2 + v \log(d^2 + \alpha).$$



Cross-entropy Method for Policy Search

Given:

- A policy $\pi_\theta(s)$ with parameters $\theta \in \mathbb{R}^d$
- An evaluation function (e.g., total reward from a rollout)
- A Gaussian distribution over parameters: $\theta \sim \mathcal{N}(\mu, \Sigma)$



Repeat for each iteration:

1. **Sample** N parameter vectors $\theta^{(1)}, \ldots, \theta^{(N)}$ from $\mathcal{N}(\mu, \Sigma)$
2. **Evaluate** each sampled policy by running it in the environment and recording its total reward
3. **Select** the top K samples with the highest rewards ("elite samples")
4. **Update** the mean μ and standard deviation Σ using only the elite samples
5. Repeat until convergence or a maximum number of iterations is reached

Experimental Results

Stability-Guaranteed Reinforcement Learning for Contact-rich Manipulation

Shahbaz A. Khader[1,2], Hang Yin[1], Pietro Falco[2] and Danica Kragic[1]

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

[1]Robotics, Perception, and Learning lab, Royal Institute of Technology, Sweden. {shahak, hyin, dani}@kth.se.

[2]ASEA Brown Boveri (ABB) Corporate Research, Sweden. pietro.falco@se.abb.com.

Correspondence to shahak@kth.se.

Stability certification tends also to reduce the number of rollouts needed

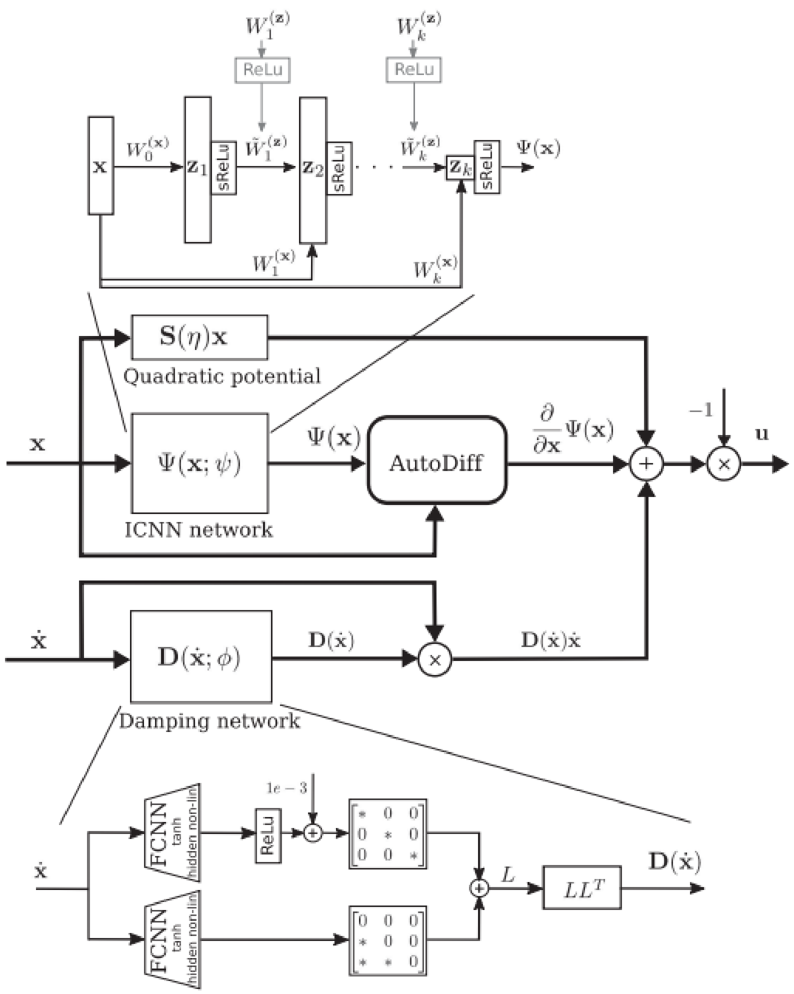
Potential Limitation

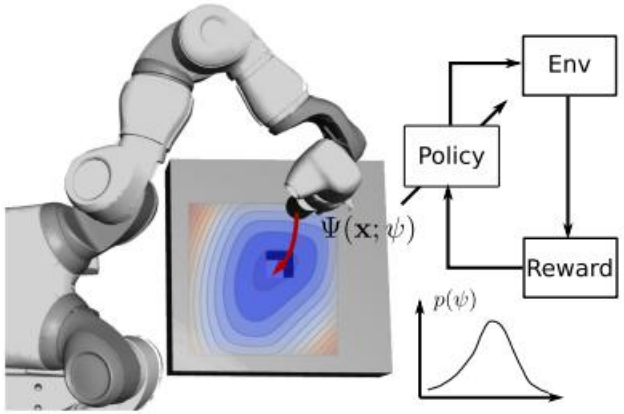
- I-Mogic has a structured policy.

$$\mathbf{u} = -\boldsymbol{S}^0 s - \boldsymbol{D}^0 \dot{s} - \sum_{k=1}^{K} w^k(s)[\boldsymbol{S}^k(s - s^k) + \boldsymbol{D}^k \dot{s}]$$

- What if we want to have more complex behaviors?
- Is it possible to use deep learning?

Energy Shaping Policy via Deep Networks





Positive definite

Semi positive definite

$$\mathbf{u} = -\mathbf{S}(\eta)\mathbf{x} - \frac{\partial}{\partial \mathbf{x}}\Psi(\mathbf{x};\psi) - \mathbf{D}(\dot{\mathbf{x}};\phi)\dot{\mathbf{x}}$$

Convex function

ICNN (Input Convex Neural Network)

- a special type of neural network where the output is **guaranteed to be convex with respect to its input**.
- Useful to learn an energy function

Quadratic Potential:

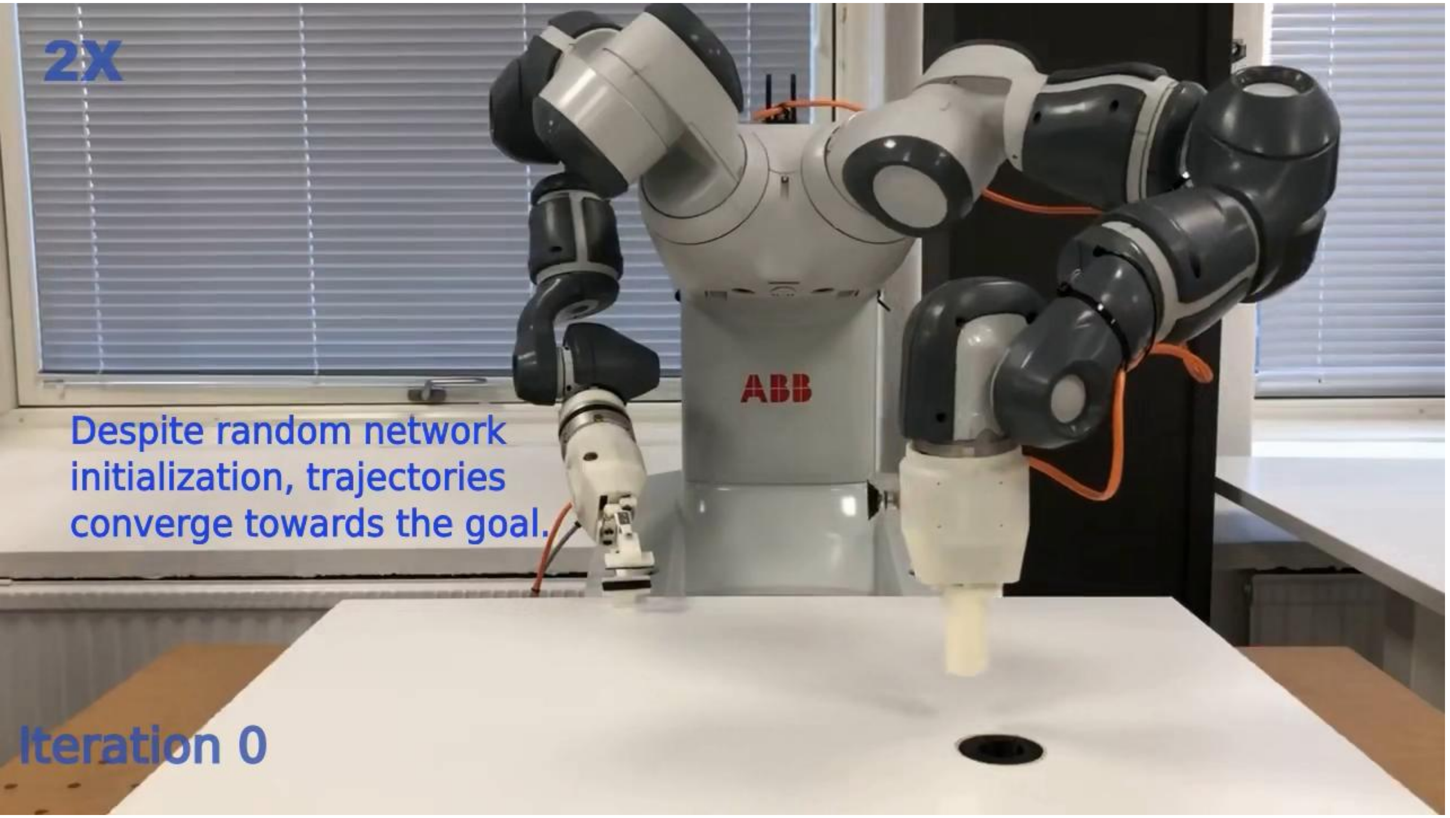
- To guarantee strong convexity

Damping (fully connected network):

- To guarantee a smooth motion, designed to be semipositive definite, ensure passivity

Khader, S. A., Yin, H., Falco, P., & Kragic, D. (2021). Learning deep energy shaping policies for stability-guaranteed manipulation. IEEE Robotics and Automation Letters, 6(4), 8583-8590.

Energy Shaping Policy



We guarantee stability all-the-time stability with a more expressive policy

The reward function is still hand-crafted bu the user. Can we specify it with natural language?

Some Key Challenges in Adaptation

1. High number of rollouts, potentially also in sim2real

   Many works in the scientific community to reduce rollouts on the real robot (model-based reinforcement learning, sim2real)

2. Irreversible events

3. Ensuring formal guarantees during the different rollouts

4. **Express reward/cost functions without having technical skills**

Next Step: Automate Reward Generation

We would like to generate rewards from human natural language

Task definition

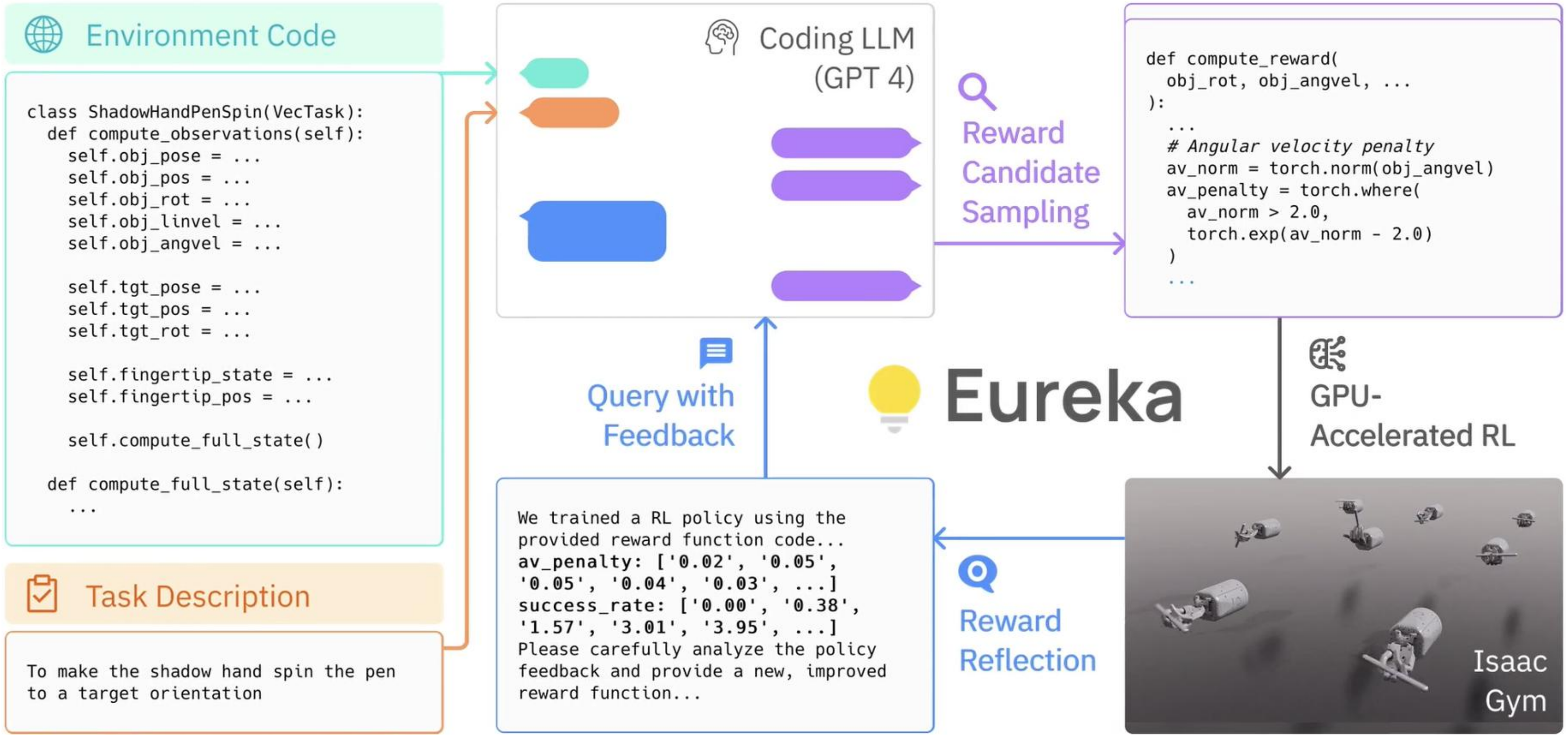
What skill should the robot learn

Reward generation

and

Agent training

Eureka Approach



1. Multiple Reward Function Generation
From taks desccription via LLM

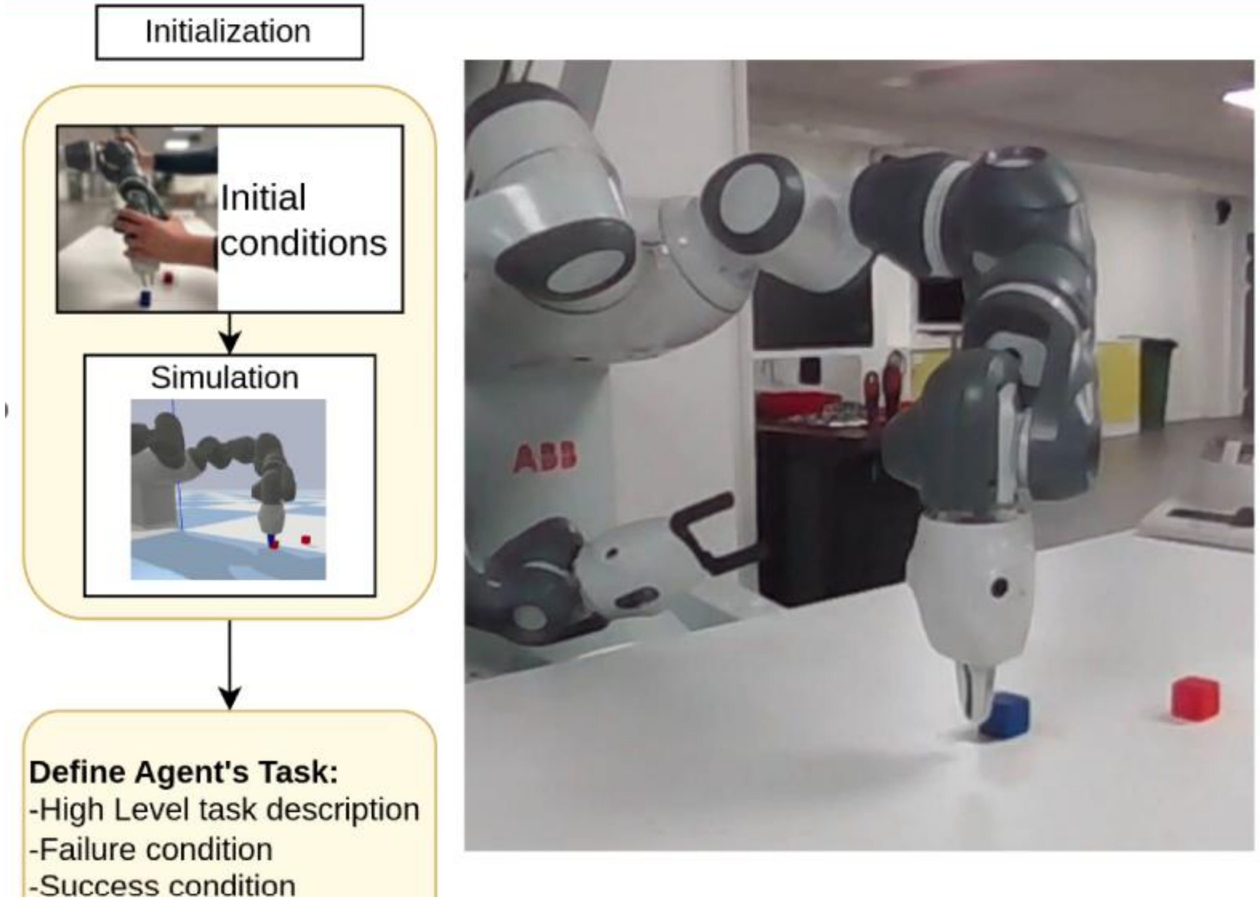
2. Evaluation in parellel on Isaac Gym

3. Results are reported back to the LLM

4. Until convergence go to step 1

Ma, Yecheng Jason, et al. "Eureka: Human-level reward design via coding large language models." *arXiv preprint arXiv:2310.12931* (2023).

Reward Based on task Description



Task description:
"The robot's gripper is close to a blue cube, touch it with the gripper fingers and push it close to the red cube."

Success condition:
"Consider the task solved it the distance between the cubes is less than 0.04 meters."

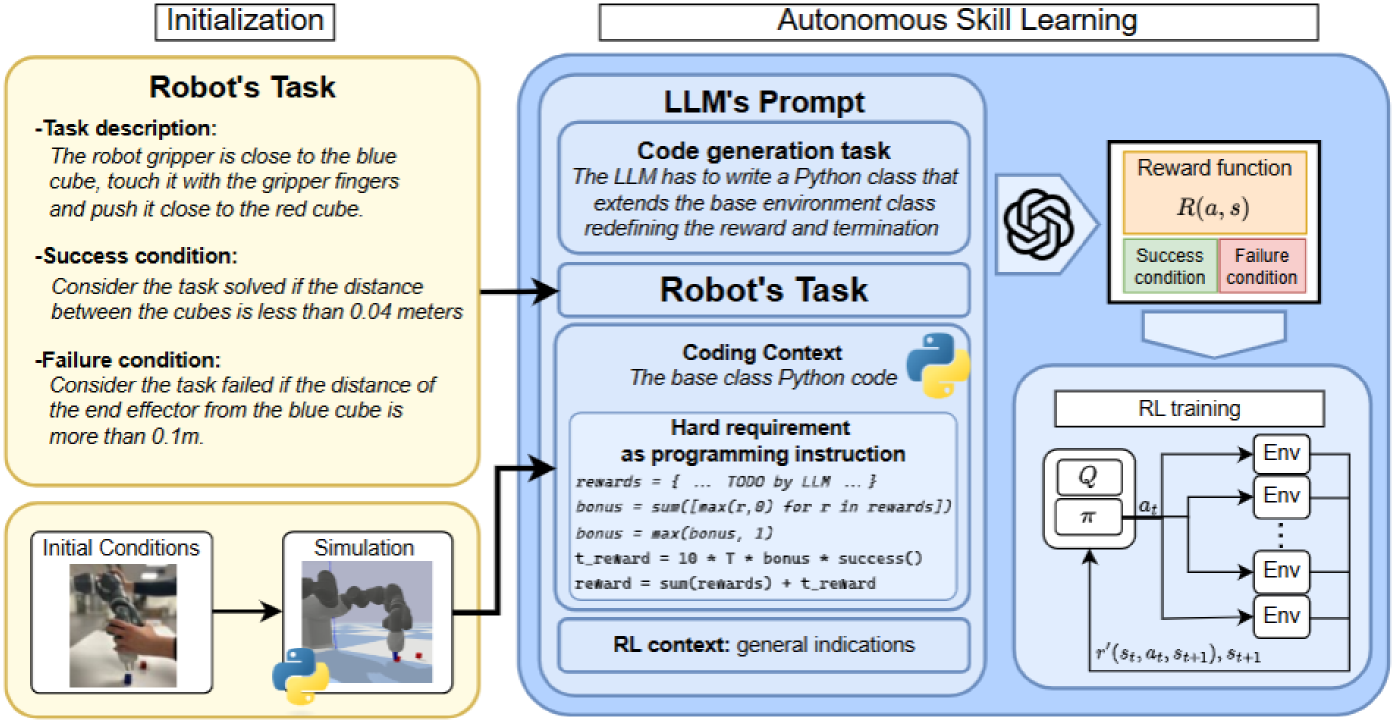
Failure condition:
"Consider the task failed if the distance between the end effector and the blue cube is more than 0.1 meters."

Turcato, N., Iovino, M., Synodinos, A., Libera, A. D., Carli, R., & Falco, P. (2025). Towards Autonomous Reinforcement Learning for Real-World Robotic Manipulation with Large Language Models. *IEEE RA-L, 2025.*

ARCHIE Approach

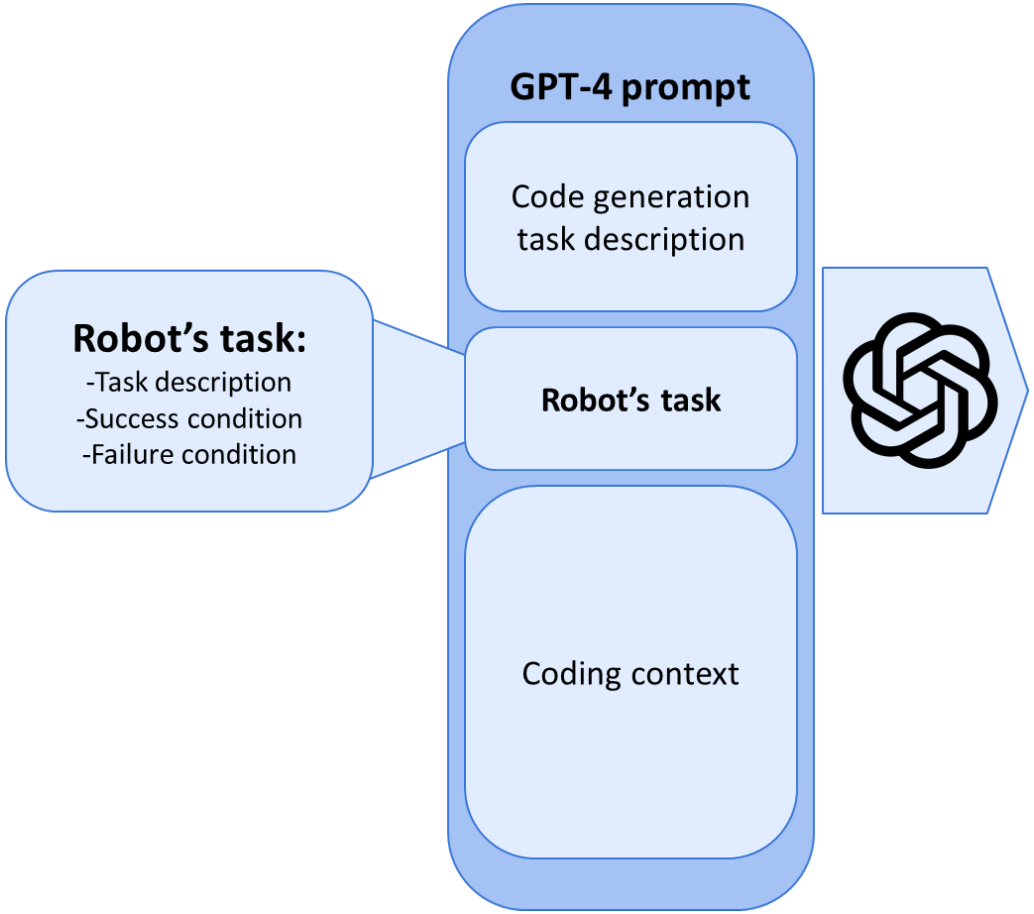
We want to learn reward functions for manipulation in one-shot

How to do that?



Turcato, N., Iovino, M., Synodinos, A., Libera, A. D., Carli, R., & Falco, P. (2025). Towards Autonomous Reinforcement Learning for Real-World Robotic Manipulation with Large Language Models. *arXiv preprint arXiv:2503.04280*.

Naive Reward Generation




```
rewards_dict = {
  "dist_blue": -dist_blue,
  "dist_red_blue": -dist_red_blue
  "contact_reward": int(finger_1) + int(finger_2)
}

reward = sum([r for r in rewards_dict.values()])
```

Rewards generated by the LLM are generally coherent with the task definition. They are generally numerically unstable, causing the agent to over-visit non-terminal states, i.e. where the task is not solved.

$$\sum_{k=1}^{K} r^k(s_t, a_t) + R_F(s_t, a_t)$$

ARCHIE: Solution for Reward Generation

We Want to guarantee that $\boxed{R_F(a_t, s_t) \gg \sum_{t=0}^{T-1} r(s_t, a_t, s_{t+1})}$ by introducing the following term:

$$r(s_t, a_t, s_{t+1}) = \underbrace{\sum_{k=1}^{K} r^k(s_t, a_t)}_{\underbrace{\sum_{k^+ \in K^+} r^{k^+}(s_t, a_t)}_{\text{Bonus (positive terms)}} + \underbrace{\sum_{k^- \in K^-} r^{k^-}(s_t, a_t)}_{\text{Malus (negative terms)}}} + \underbrace{R_F(a_t, s_t)}_{R_F(a_t, s_t) = 10T \max\left(\sum_{k^+ \in K^+} r^{k^+}(s_t, a_t), 1\right)} \overbrace{\Phi(s_{t+1})}^{\Phi(s_{t+1}) = 1 \text{ if task is solved in } s_{t+1},\ 0 \text{ otherwise}}$$

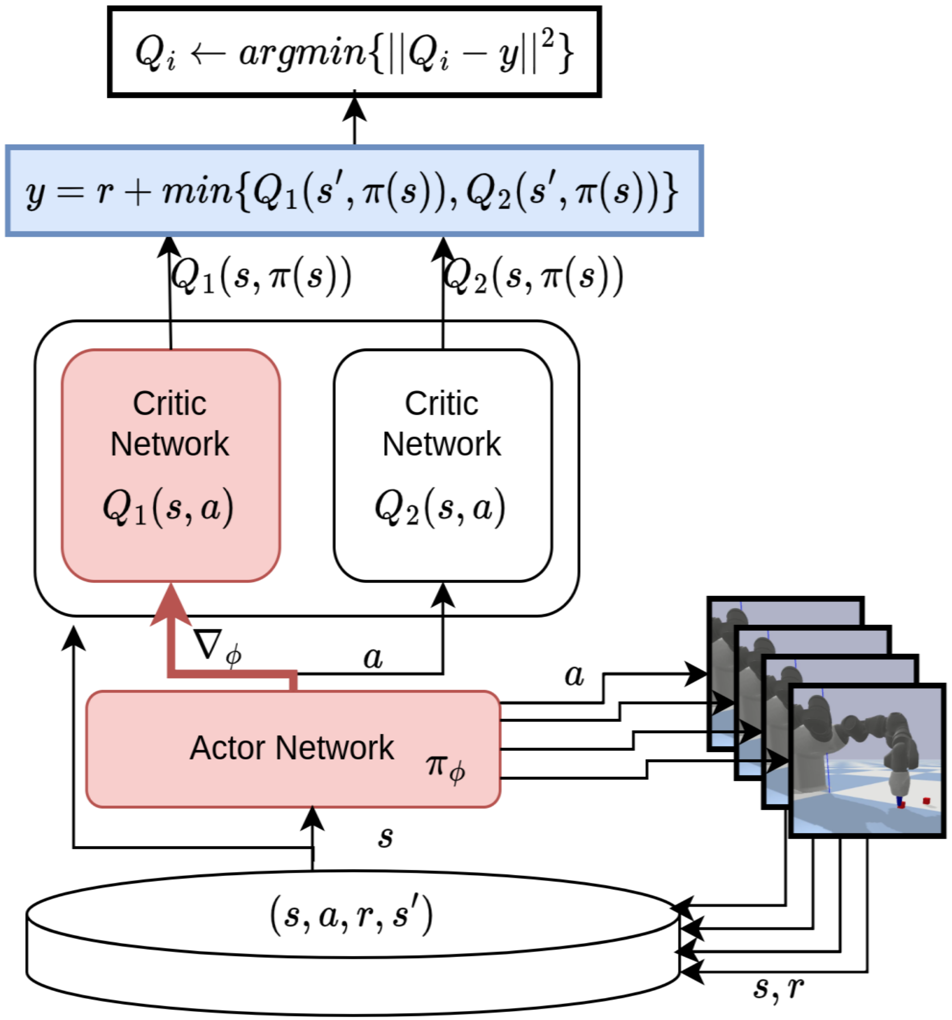
$$\boxed{R_F(a_t, s_t) \gg \sum_{t=0}^{T-1} r(s_t, a_t, s_{t+1})}$$

Turcato, N., Iovino, M., Synodinos, A., Libera, A. D., Carli, R., & Falco, P. (2025). Towards Autonomous Reinforcement Learning for Real-World Robotic Manipulation with Large Language Models. *arXiv preprint arXiv:2503.04280*.

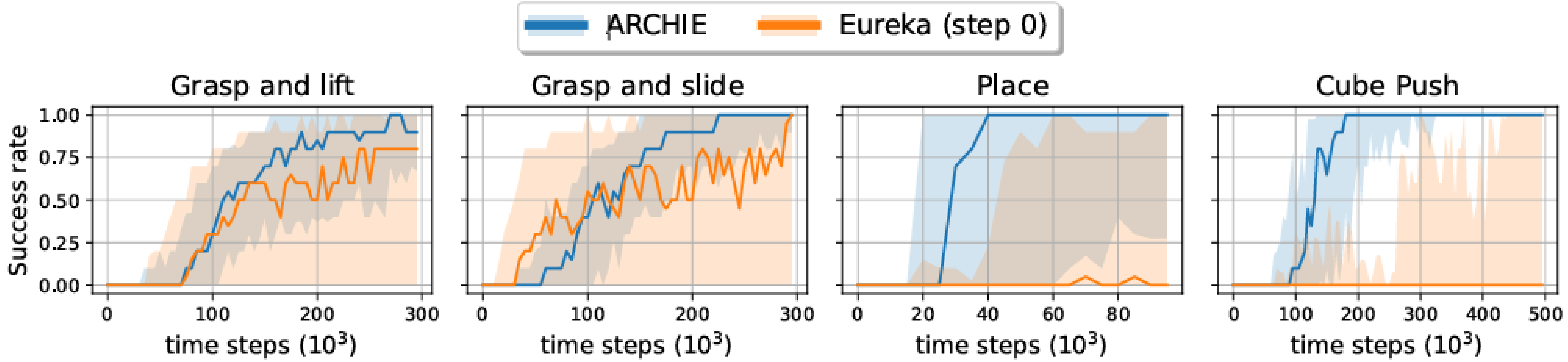
Autonomous skill learning: Agent training

Soft Actor Critic, with parallel exploration:

- Stochastic Off-Policy Deep Reinforcement Learning algorithm;
- Very popular in robotics, due to the stochastic policy exploration properties.



Results


ARCHIE
Eureka (step 0)
Grasp and lift
Grasp and slide
Place
Cube Push
Success rate
1.00
0.75
0.50
0.25
0.00
0
100
200
300
20
40
60
80
400
500
time steps (10^3)

Comparison of the two approaches

Aspect	ARCHIE	Eureka
Approach	**One-shot** reward generation via GPT-4	**Iterative** reward refinement via GPT-4 + reward feedback loop
LLM usage	Generates reward code from natural language once	Generates multiple reward candidates, refined through in-context updates
Refinement loop	**None:** single generation	**Yes,** guided by RL training outcomes
Reward structure	Structured: shaping + terminal reward.	Arbitrary executable Python reward functions
Simulation backend	PyBullet, Mujoco, test on real ABB YuMi	Isaac Gym (massively parallel GPU RL)
RL algorithm	SAC (continuous control)	Evolutionary + PPO
Target domain	Industrial manipulation, sim2real	Dexterous manipulation, locomotion, curriculum learning

Take-Home Message

- Learning alone is often not enough in real worlds— formal guarantees and robustness to irreversible events are important in human-centered robotics.

- We need robots that have a good trade-off between predictability, formal guarantees, and exploration.

- Using the power of transformer-based technologies can potentially allow robot to learn based on feedback and communication with humans

Acknowledgement








Thank you