TNG TECHNOLOGY CONSULTING

General Purpose Robotics with ROS

Tobias Weyer
Daniel Klingmann

2025-11-19

Straßburg
ROSCon FR & DE 2025

Speakers

Tobias Weyer

tobias.weyer@tngtech.com

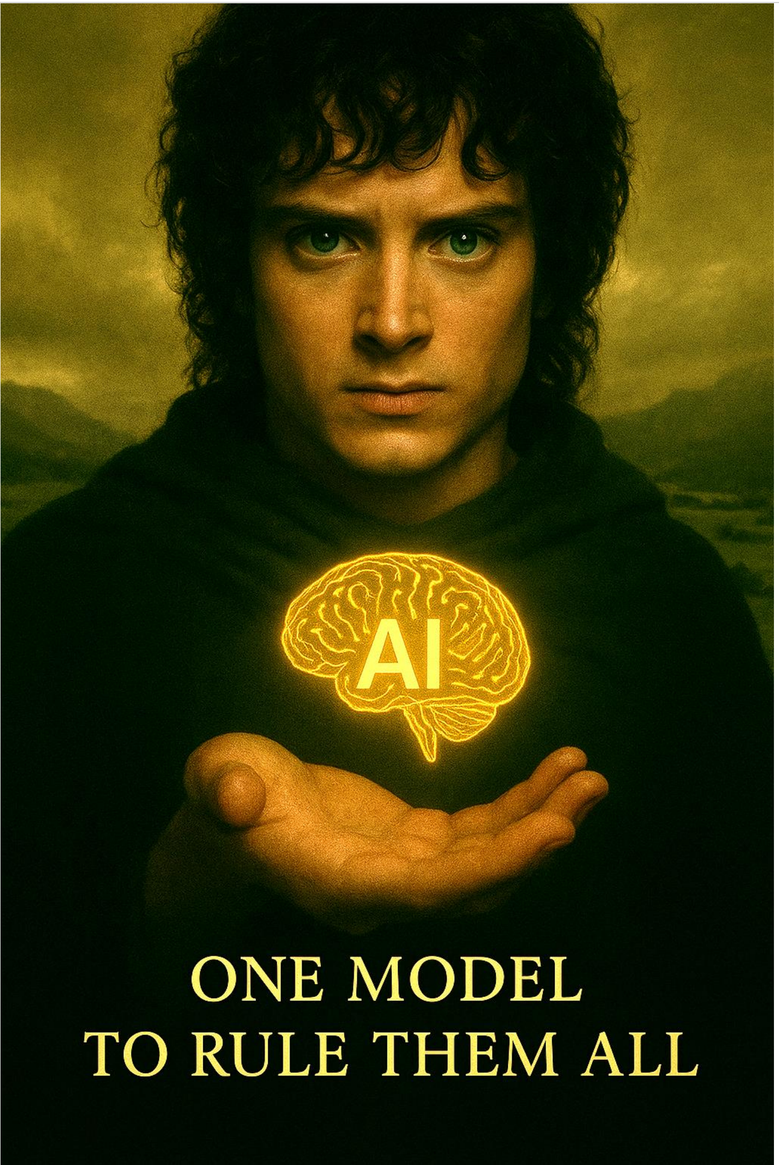
- Software Consultants at TNG Technology Consulting in Munich
- Background in Mechatronics, IT and Mathematics
- Researching Vision-Language-Action Models

Daniel Klingmann

daniel.klingmann@tngtech.com

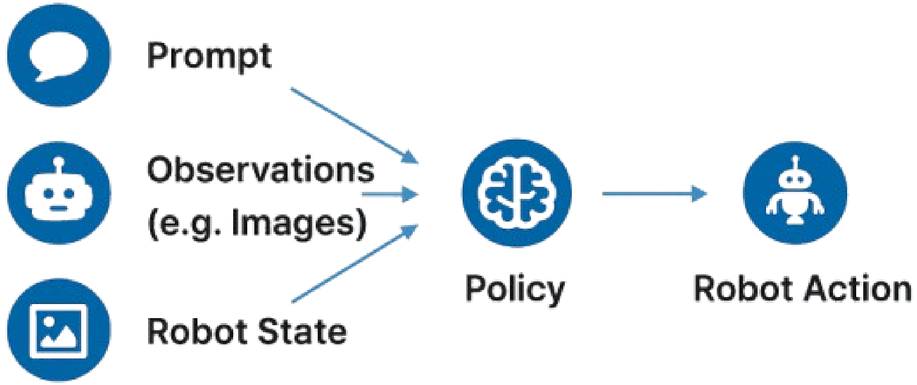
What is General Purpose Robotics

What is General Purpose Robotics



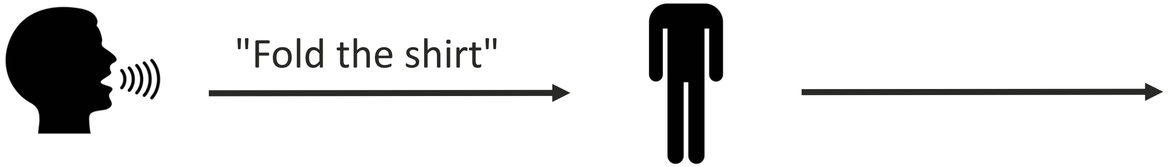
One Policy

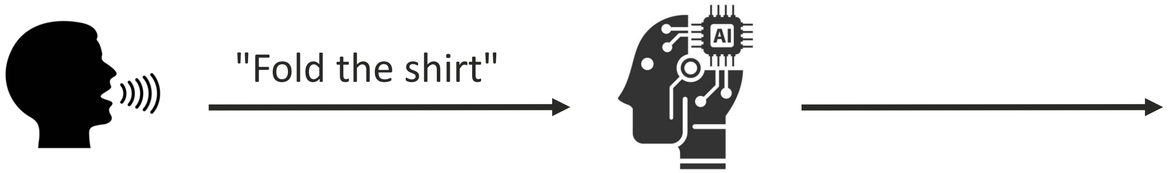
- For complex and dexterous tasks
- For different tasks
- For different embodiments



What is General Purpose Robotics

Onboarding like a human





https://www.physicalintelligence.company/blog/pi0

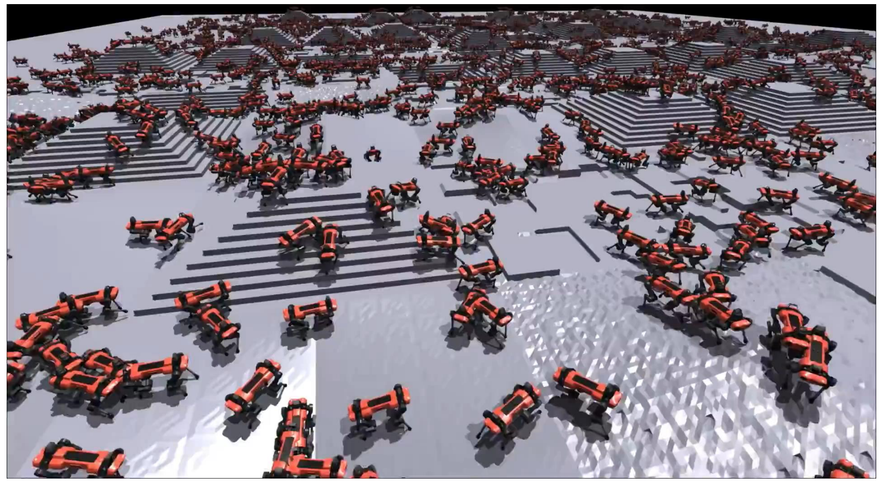
What is General Purpose Robotics

TNG TECHNOLOGY CONSULTING

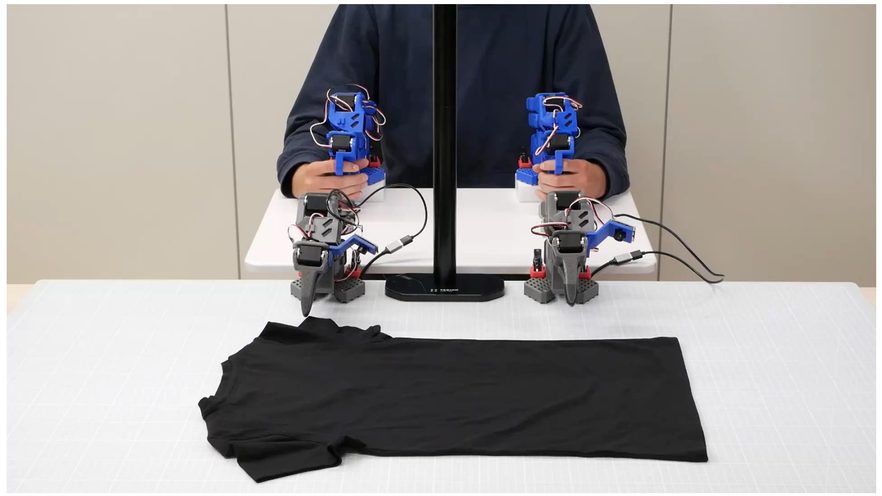
Practice makes the master

- **Reinforcement Learning**
  - Automated try and error
  - (almost) only applicable in simulation

- **Immitation Learning**
  - Learn from recorded training data
  - Lot of data needed



https://leggedrobotics.github.io/legged_gym/



Vision Language Models

Large Language and Vision Assistant

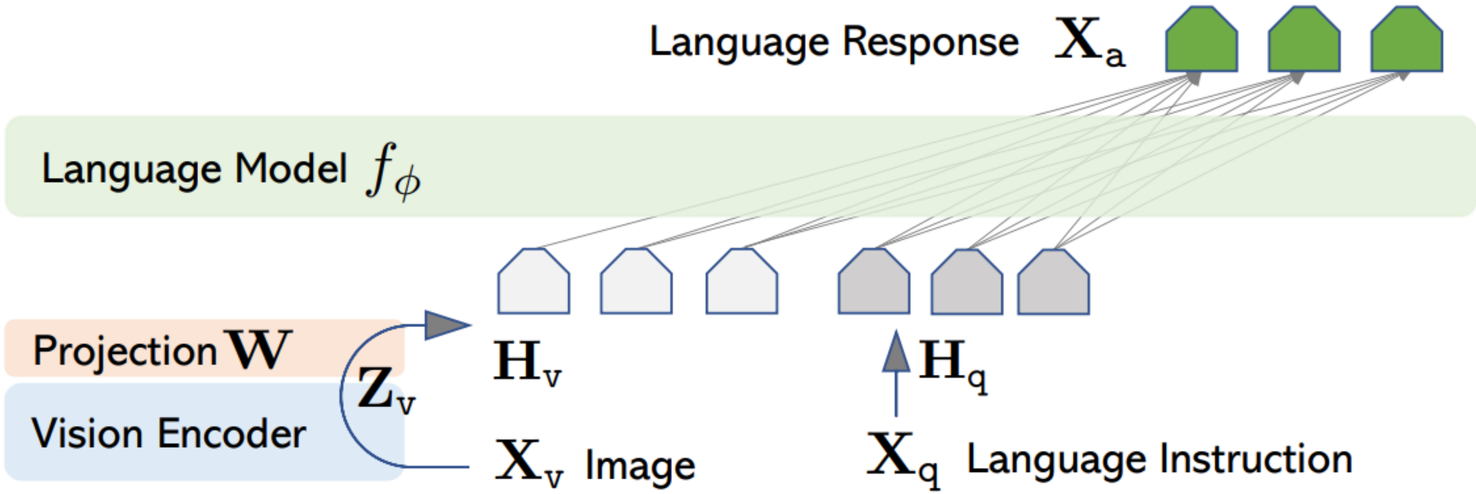
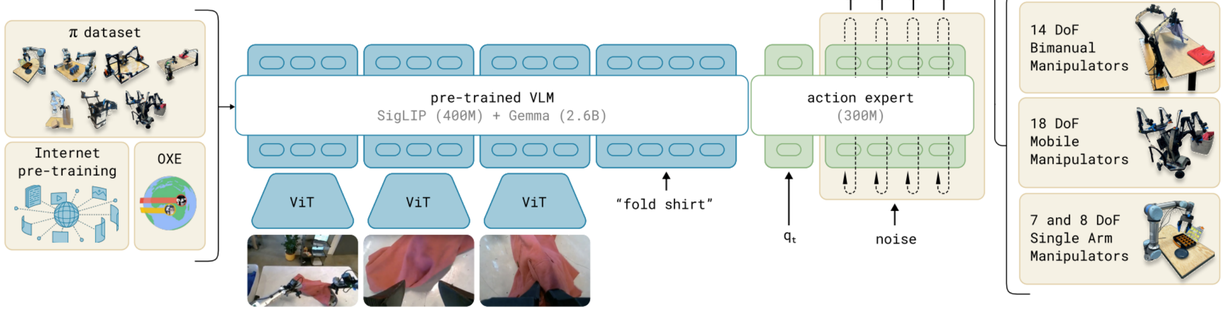


Figure 1: LLaVA network architecture.

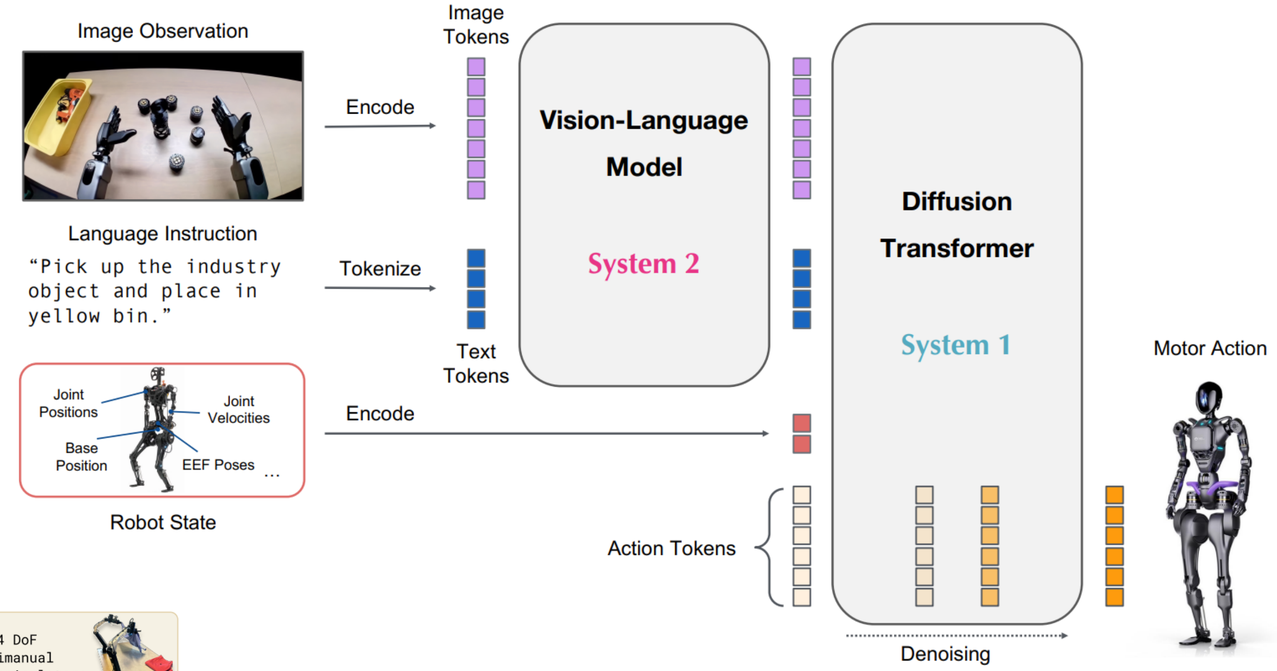
https://arxiv.org/pdf/2304.08485
Figure 1

Vision Language Action Models

PI0 and GR00T



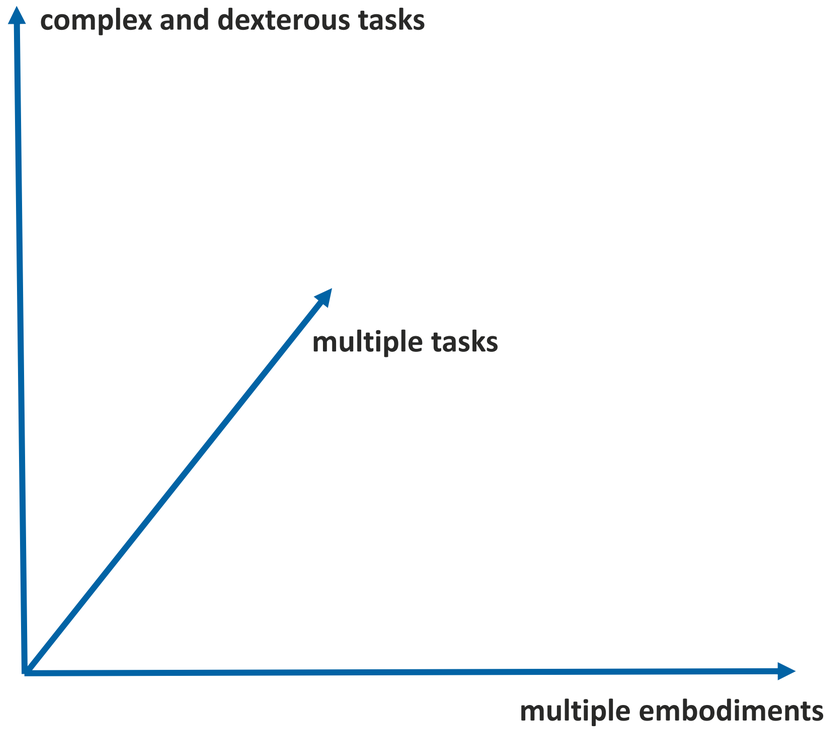
PI0: https://arxiv.org/pdf/2410.24164
Figure 3

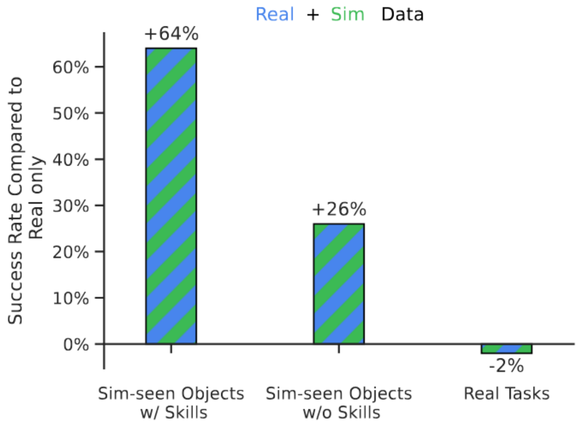


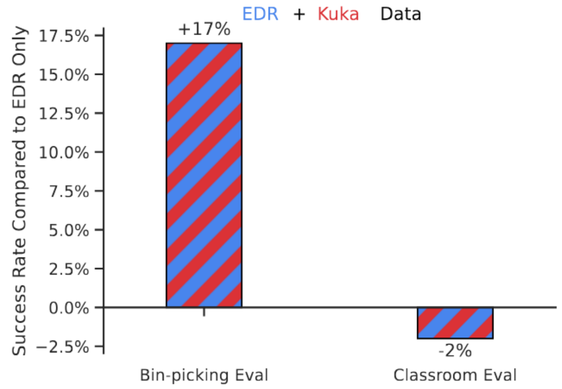
GR00T: https://arxiv.org/pdf/2503.14734
Figure 2

Vision Language Action Models

3 Dimensions of Generalization





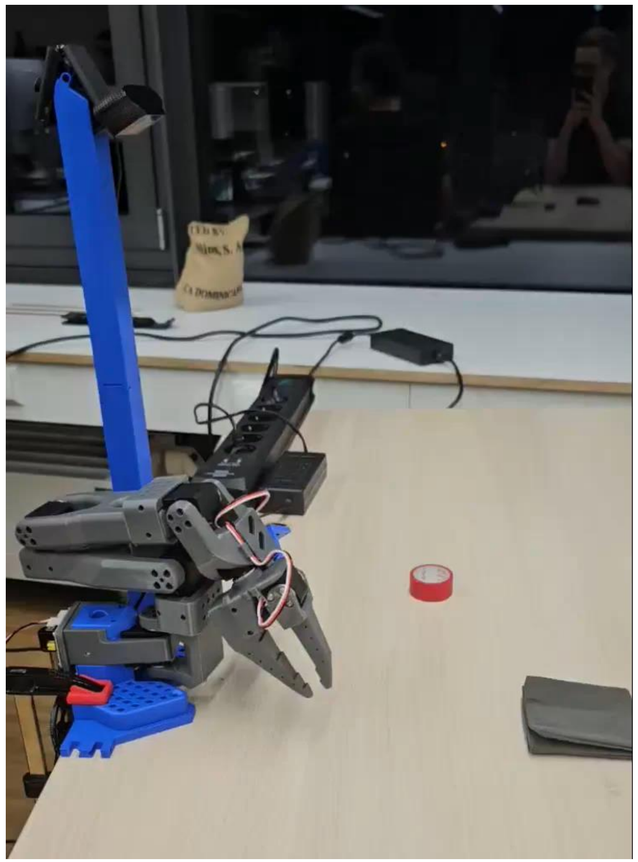


Brohan, A. *et al.* (2023). *RT-1: Robotics Transformer for Real-World Control at Scale*. arXiv:2212.06817.

Experiences from TNG

TNG TECHNOLOGY CONSULTING

Early Approaches



Task specific finetuning required

Hard to reproduce results

Hard to benchmark

Data collection and policy evaluation is very time consuming

Experiences from TNG

Sim to Real

Setup:

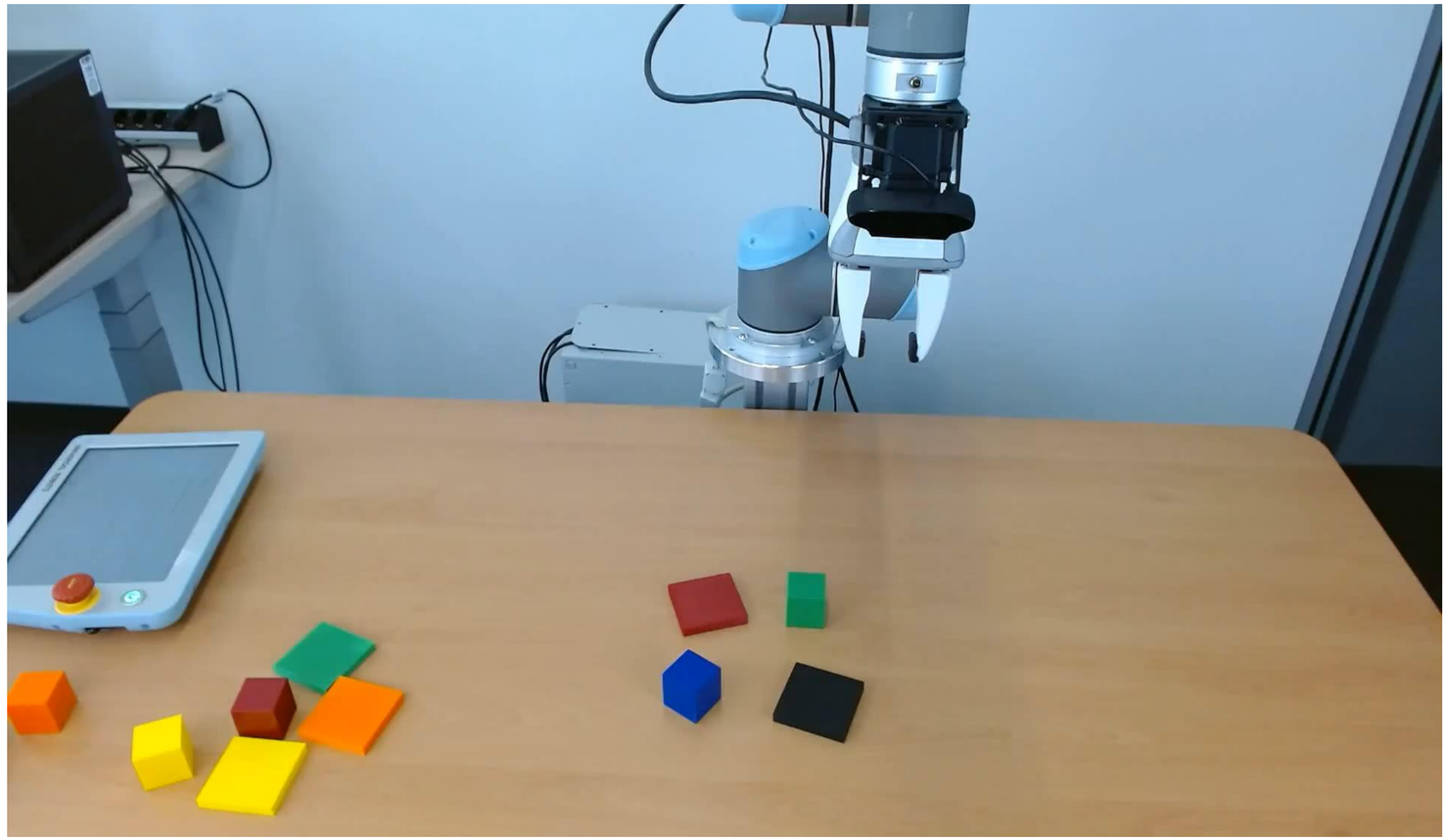
- GR00T N1.5
- Trained with simulation data from Isaac Lab
- 300 simulated episodes
- Prompt: *"Pick up the <color> cube and place it on the <color> platform"*

Experiences from TNG

Sim to Real

Evaluation:

- 100% success rate in sim
- Deploymed on real robot (70% success rate)
- Sensitive to lighting conditions
- Strong out of distribution generalization
  - Different surrounding than simulation
  - Worked for colors that weren't in training data
  - Worked with different prompts in different languages (e.g. *"Bitte, lieber Roboter, hebe den himmelfarbenen Würfel auf und setze ihn auf die bewölkte Platform, Danke"*)



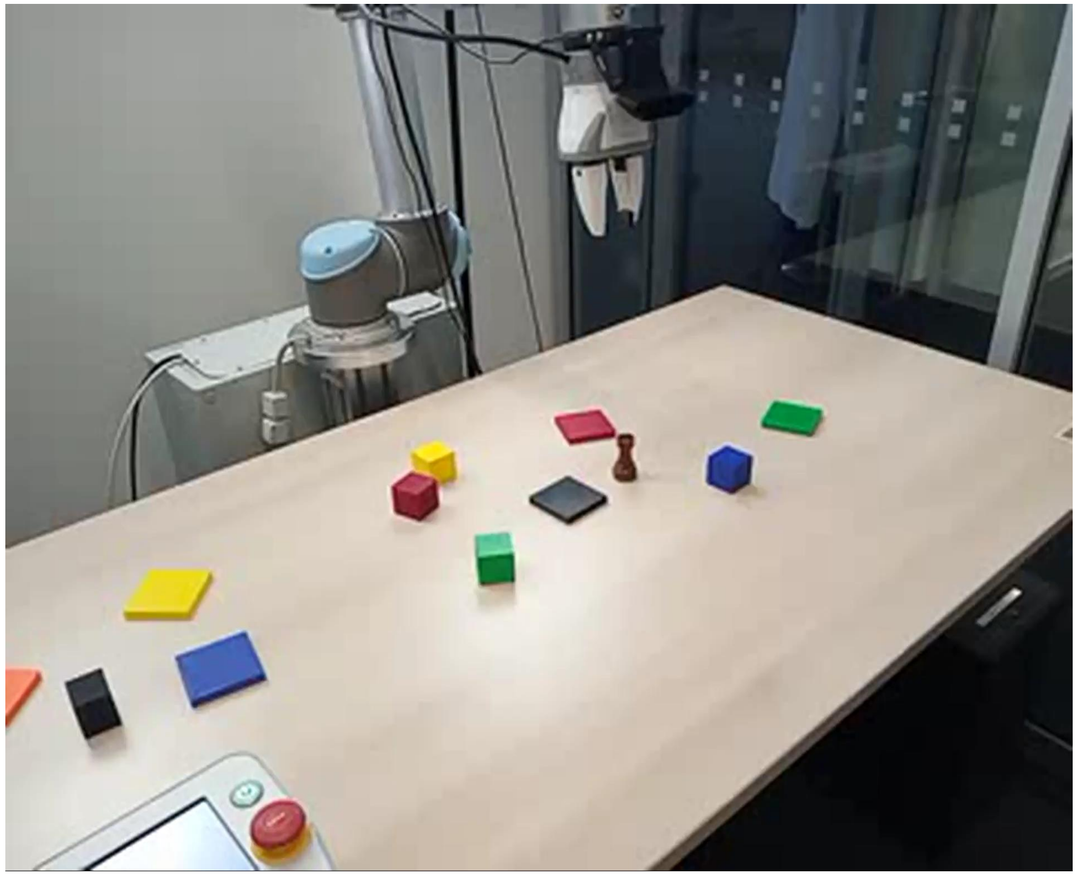
Experiences from TNG

Sim to Real

Challenges:

- Overfitted on kinematic space of the training data

  -> Robot recognizes the chess tower but is unable to close the gripper far enough
- Adding real robot teleoperation data improves the performance for the chess tower but lowered the success rate for cubes



Prompt: *"Pick up the tower and place it on the black platform"*

Experiences from TNG

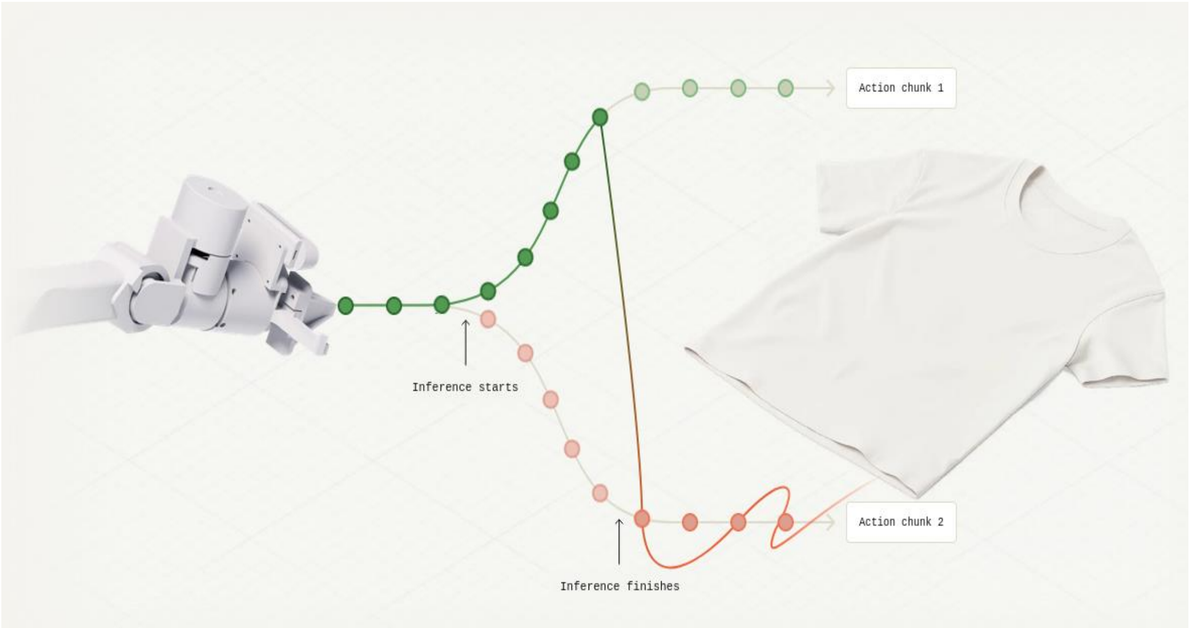
Real Time Action Chunking

Problem:

- Synchronous inference and action execution leads to jerky movements

Proposed Solution by Physical Intelligence:

- Real-Time-Action-Chunking (RTC)
- "Inpainting" of next action chunk like for images



https://www.physicalintelligence.company/research/real_time_chunking

Experiences from TNG

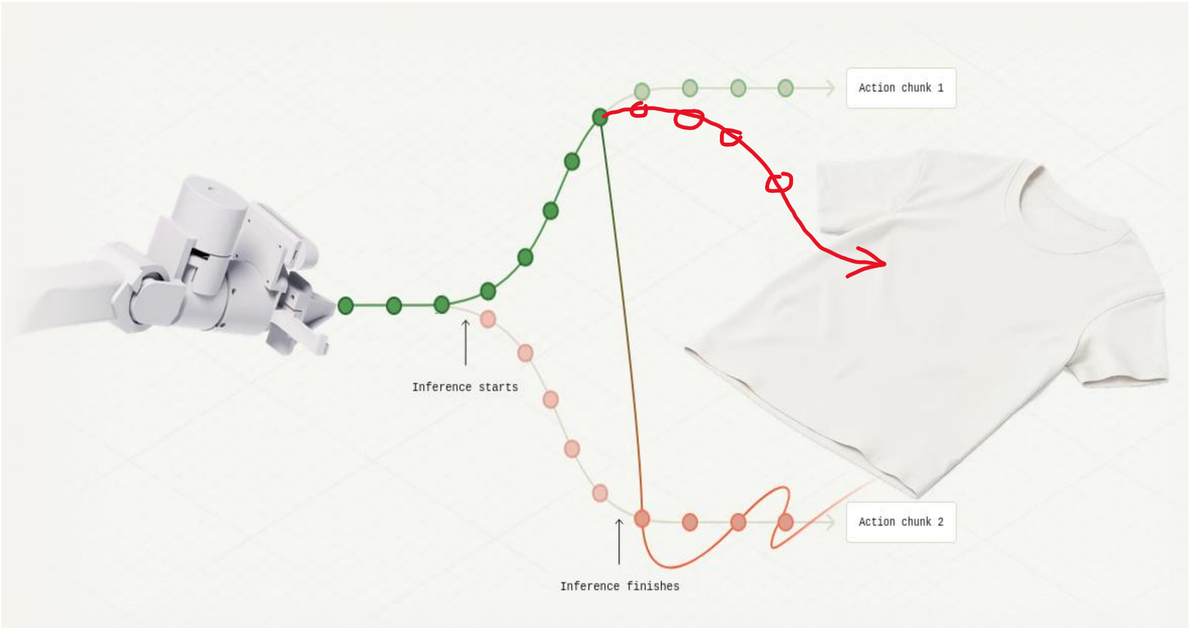
Real Time Action Chunking

Problem:

- Synchronous inference and action execution leads to jerky movements

Proposed Solution by Physical Intelligence:

- Real-Time-Action-Chunking (RTC)
- "Inpainting" of next action chunk like for images



https://www.physicalintelligence.company/research/real_time_chunking

Experiences from TNG

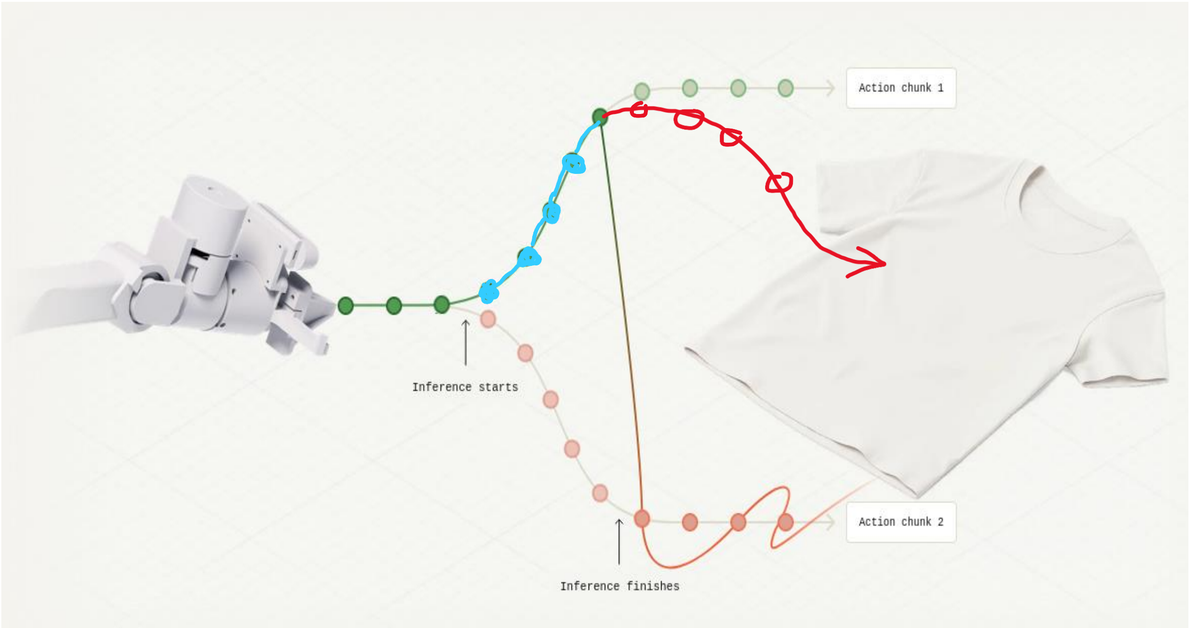
Real Time Action Chunking

Problem:

- Synchronous inference and action execution leads to jerky movements

Proposed Solution by Physical Intelligence:

- Real-Time-Action-Chunking (RTC)
- "Inpainting" of next action chunk like for images



https://www.physicalintelligence.company/research/real_time_chunking

Experiences from TNG

Real Time Action Chunking

Problem:

- Synchronous inference and action execution leads to jerky movements

Proposed Solution by Physical Intelligence:

- Real-Time-Action-Chunking (RTC)
- "Inpainting" of next action chunk like for images

Experiences from TNG

Further general learnings

finetuning with delta joint actions or relative actions works better because the possible move directions are easier to learn than the whole robot configuration space

more cameras bring only more value when they actually see more (e.g. when the robot body is blocking the view)

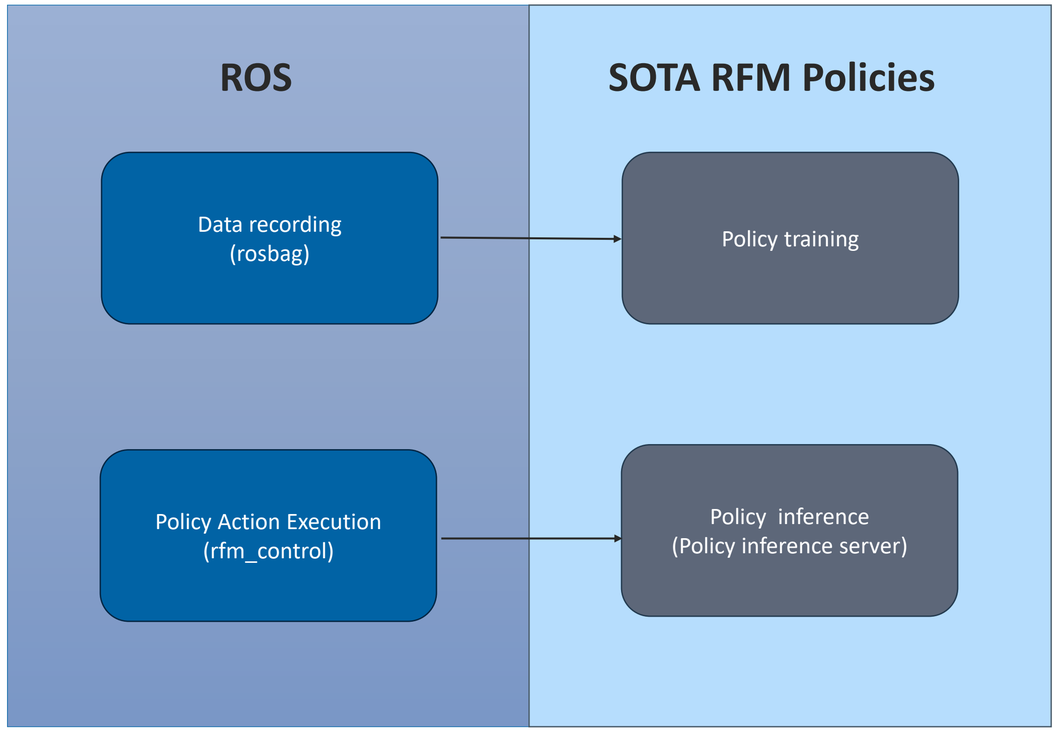
hard to benchmark in real world because of all kind of varying environment impacts

evaluation is very time consuming

VLA integration with ROS

How does ROS help?

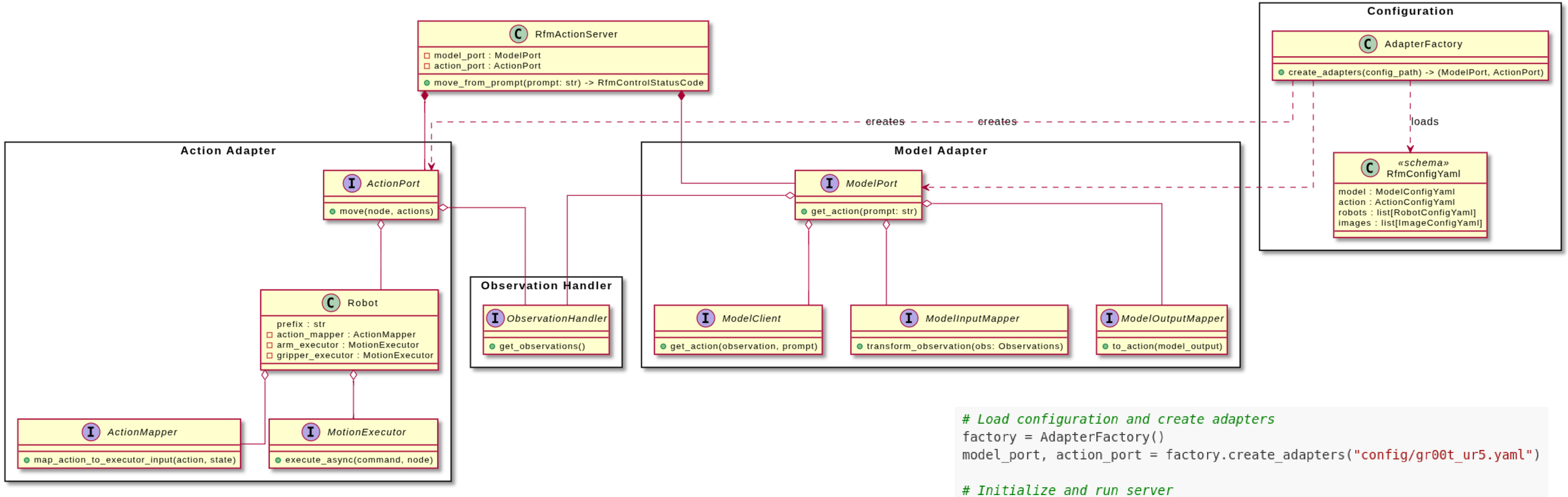
- Easy data recording with rosbag
- Great visualization capabilities out of the box (Rviz)
- Simulation interfaces
- One implementation for different robot models



VLA integration with ROS

TNG TECHNOLOGY CONSULTING

Open source rfm_control package




```python
# Load configuration and create adapters
factory = AdapterFactory()
model_port, action_port = factory.create_adapters("config/gr00t_ur5.yaml")

# Initialize and run server
server = RfmActionServer(model_port, action_port)
server.move_from_prompt("pick up the cup")
```

VLA integration with ROS

Open source rfm_control package

- **Configuration-Driven**: Change robot setup, model type, or control strategy by editing YAML files
- **Hexagonal Architecture:** Easy to swap real implementations with mocks
- **Extendable AdapterFactory:** Simply add your own policy or action adapter
- **Documentation:** Extensive user guide

Try it out and give feedback :)

GitHub

https://github.com/NeuroTo/rfm_control

TNG TECHNOLOGY CONSULTING

Thank you for your attention

Tobias Weyer
Senior Software Consultant
tobias.weyer@tngtech.com

Daniel Klingmann
Senior Software Consultant
daniel.klingmann@tngtech.com