Sim2real Multimodal Learning for Dexterous Gripper

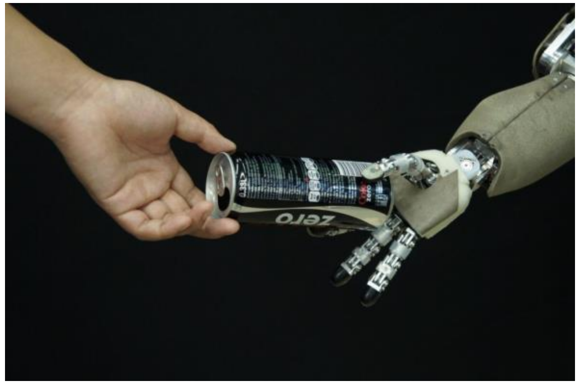
Hui Zhang
09.01.2024

Sim2real Multimodal Learning for Dexterous Gripper

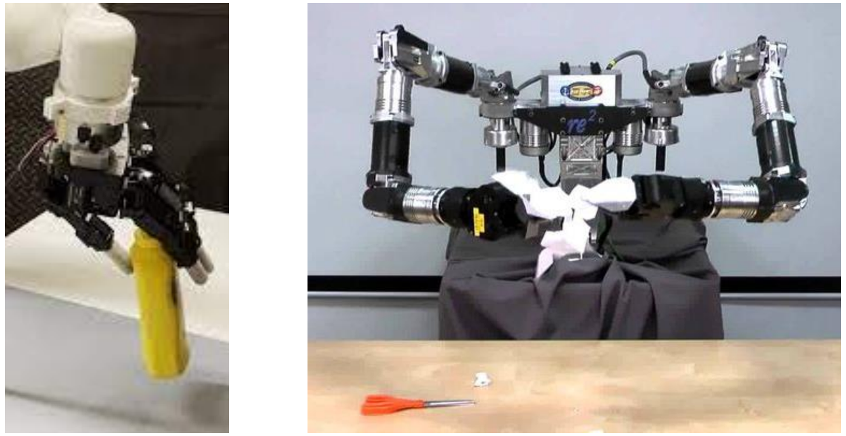
Sim2real Multimodal Learning for Dexterous Gripper

Sim2real Multimodal Learning for Dexterous Gripper

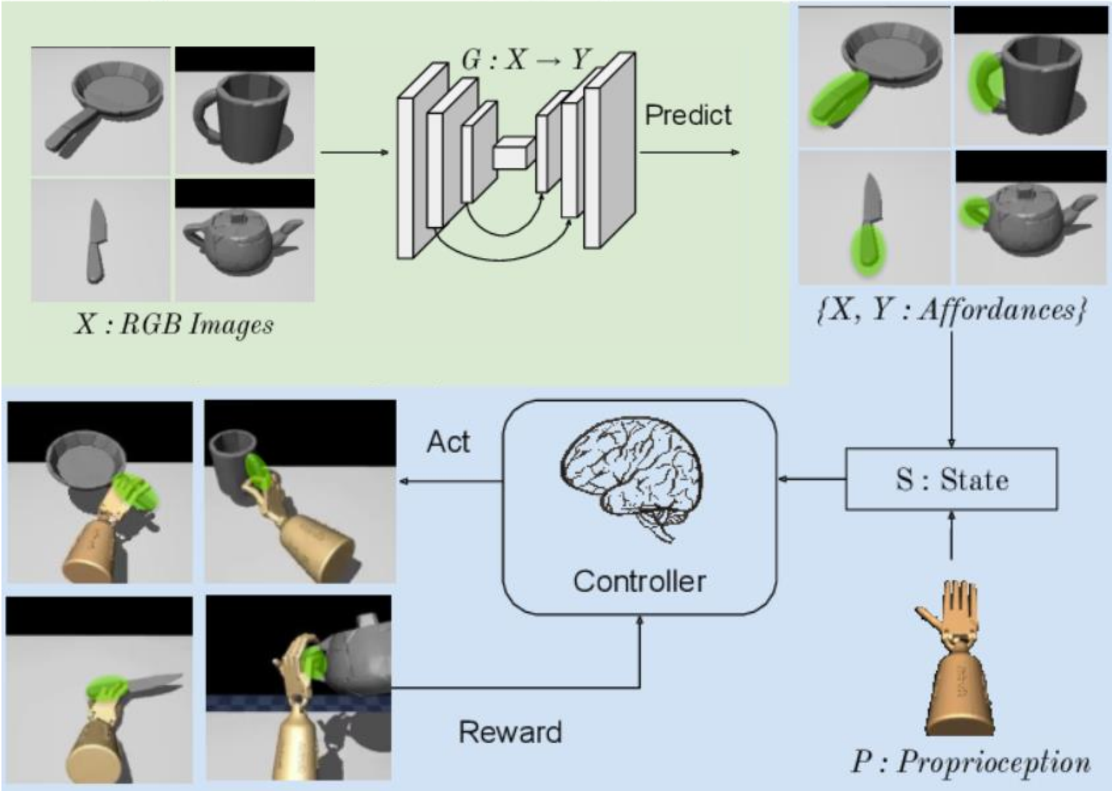
1. Research background



Assistant robot



Dexterous grasping and manipulation

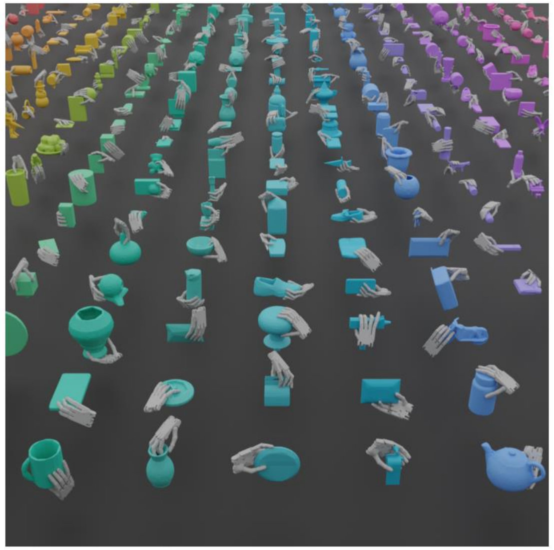


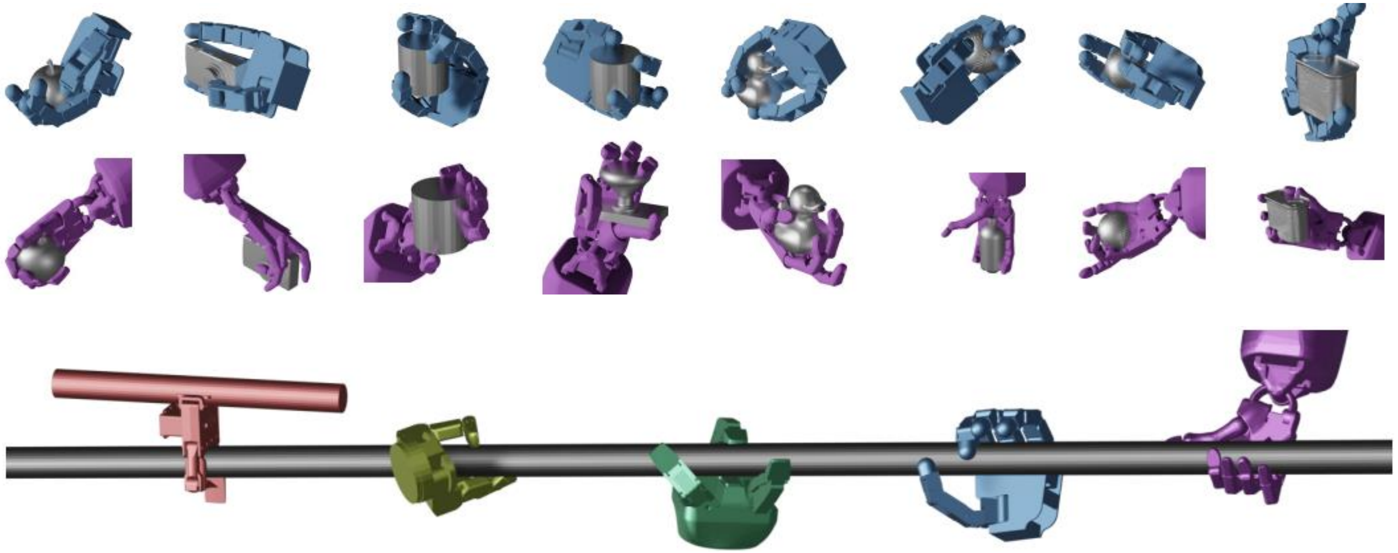
Sim2real learning for a grasping task
using a dexterous gripper

Sim2real Multimodal Learning for Dexterous Gripper

1. Research background

1.1 State of the art





Open-source datasets, containing millions of grasp examples.

[1] Li, P., Liu, T., Li, Y., Geng, Y., Zhu, Y., Yang, Y. and Huang, S. Gendexgrasp: Generalizable dexterous grasping. IEEE ICRA 2023.
[2] Wang, R., Zhang, J., Chen, J., Xu, Y., Li, P., Liu, T. and Wang, H. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. IEEE ICRA 2023.
[3] Liu, T., Liu, Z., Jiao, Z., Zhu, Y. and Zhu, S.C., 2021. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. IEEE RA-L.

Sim2real Multimodal Learning for Dexterous Gripper

1. Research background

   1.1 State of the art
   1.2 Motivation

2. Method

   2.1 Two-stage simulation and dataset collection
   2.2 Deep learning

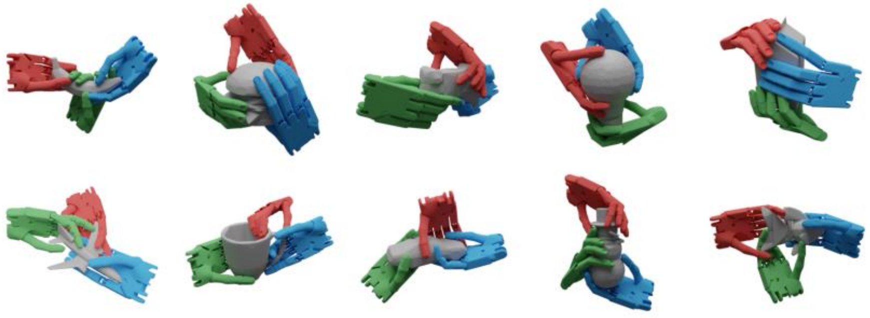
3. Conclusion

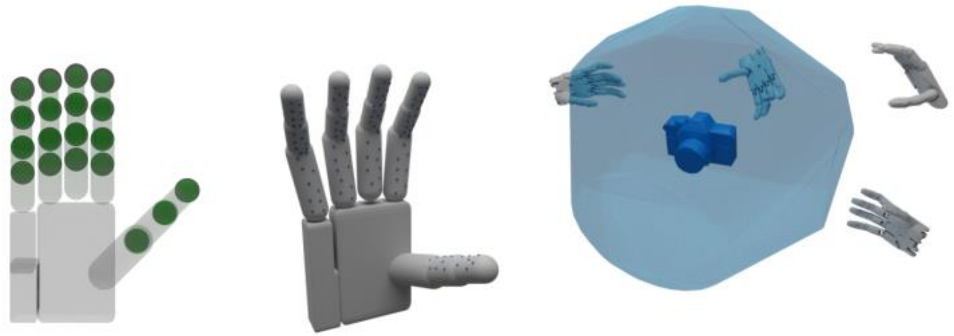
   3.1 Simulation
   3.2 NN learning
   3.3 Isaac Gym

Sim2real Multimodal Learning for Dexterous Gripper

1. Research background

1.2 Motivation

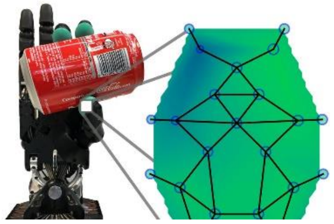




- Sim2real gaps in SOTA methods/datasets:

1) Many open-source methods/datasets generate dexterous grasp, ignoring **some** DOF ranges of real-world robotic setups -> extra grasp polices are required when deployed to robotic setups.

2) Only consider several fixed contact points on the gripper with rigid-body shapes -> not accurate for grasping and tactile sensing simulation.
   -> How many contact points should be defined to fully cover the rigid-body hand
   -> How to simulate tactile sensing for soft sensors

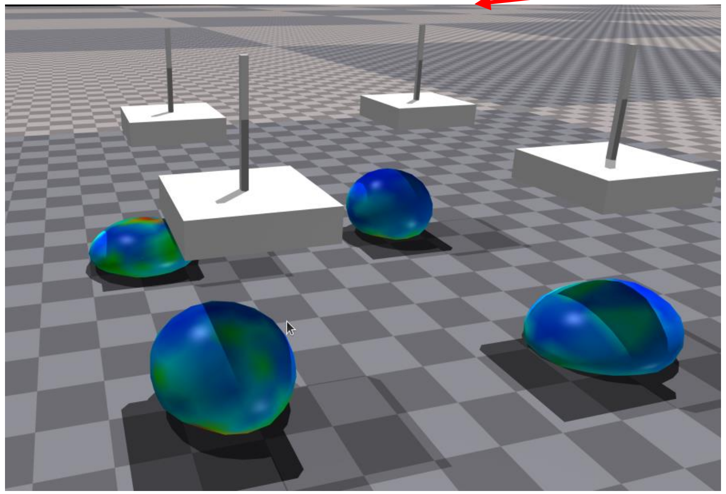


3) Not every joint can freely move for a real-world gripper.
   -> Shawdow hand -> some joints can only move together.

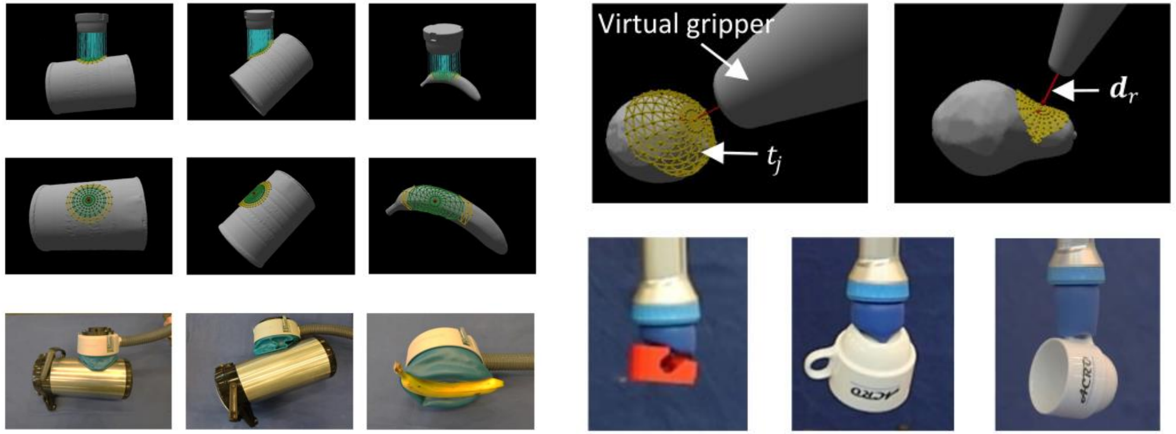
A Research Plan in My Previous Presentation

- Scenario rendering

- Two-stage contact
  Fast contact for rigid bodies (fingers and object) → soft contact (fingertips, tactile sensing)



Isaac Gym



My proposed simulation method (2022)

[1] Zhang, H., Peeters, J., Demeester, E. and Kellens, K., 2022. Deep Learning Reactive Robotic Grasping With a Versatile Vacuum Gripper. IEEE Transactions on Robotics, 39(2), pp.1244-1259.
[2] Zhang, H., Wu, Y., Demeester, E. and Kellens, K., 2022. BIG-Net: Deep Learning for Grasping With a Bio-Inspired Soft Gripper. IEEE Robotics and Automation Letters, 8(2), pp.584-591.

A Research Plan in My Previous Presentation

- 10.2023 – 11.2023

  Simulation and dataset collection

- 12.2023 – 01.2024

  1) Grasp learning
  2) Real world experiments
  3) Seminar (09.10.2024)

- 02.2023 – 03.2024

  Paper submission

Sim2real Multimodal Learning for Dexterous Gripper

1. Research background

   1.1 State of the art
   1.2 Motivation

2. Method

   2.1 Two-stage simulation and dataset collection
   2.2 Deep learning

3. Conclusion

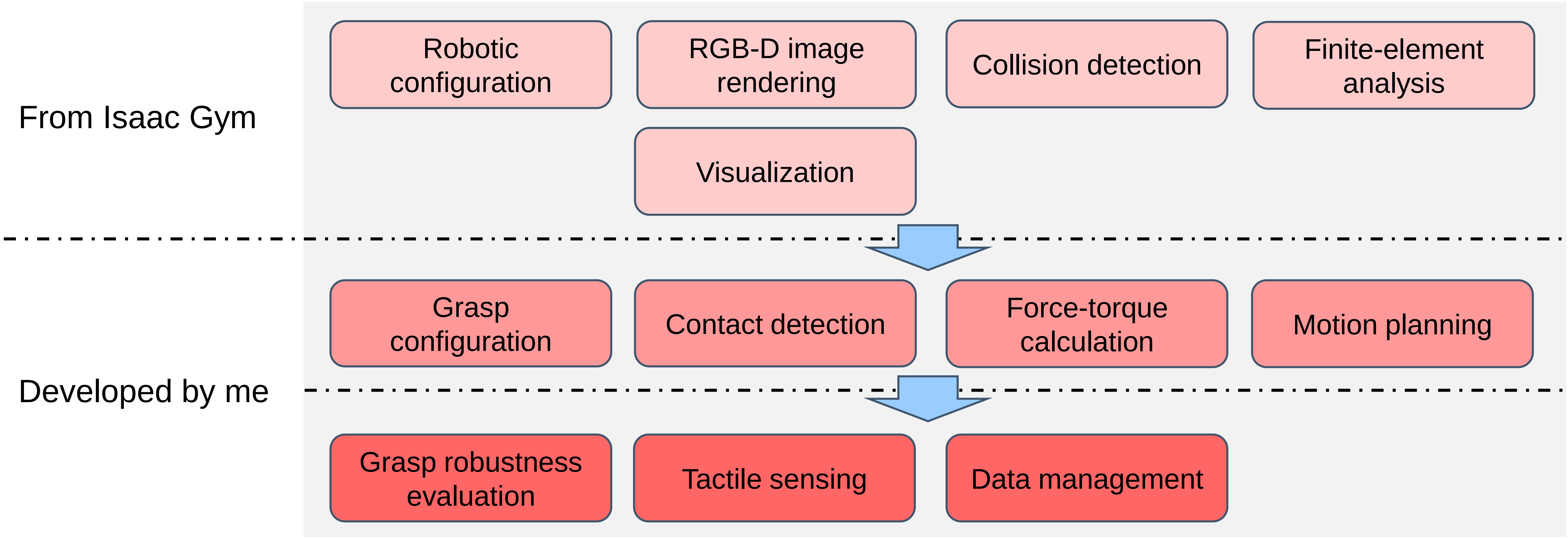
   3.1 Simulation
   3.2 NN learning
   3.3 Isaac Gym

Sim2real Multimodal Learning for Dexterous Gripper

2. Method

2.1 Two-stage simulation and dataset collection (based on Isaac Gym)

-> Frist stage: fast contact with rigid body
-> Second stage: grasp fine-tune and tactile sensing with soft body (FEM analysis)



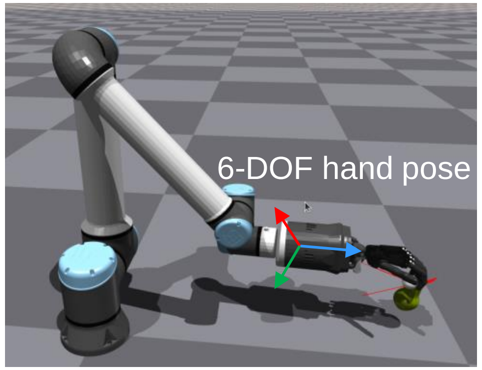
Pluggable simulation framework based on Isaac Gym

Sim2real Multimodal Learning for Dexterous Gripper

2.1 Two-stage simulation and dataset collection

-> Frist stage: fast contact with rigid body

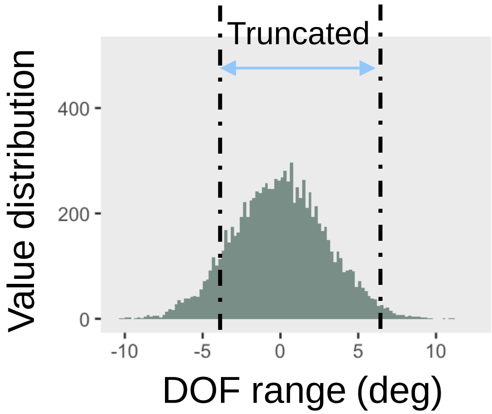
A 6-DOF hand pose:



-> random pose limited by DOF ranges of the real-world robot (UR10e + Shadowhand)

A 22-DOF fingers poses:

1) Define DOF ranges of the gripper.
2) Generate DOF targets using truncated normal distribution.
3) Randomly select a set of 22-DOF targets
4) Animations and grasp robustness evaluation

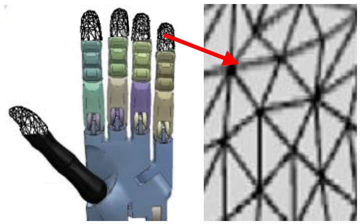


Sim2real Multimodal Learning for Dexterous Gripper

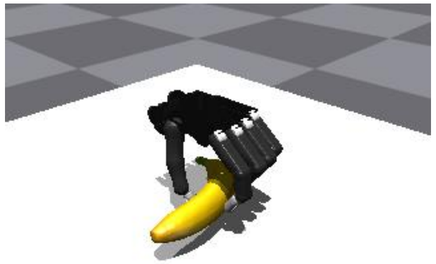
2.1 Two-stage simulation and dataset collection

-> Frist stage: fast contact with rigid body

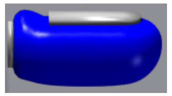
Some joints should move together.



Randomly select a set of 22-DOF targets -> contact object



Object with random pose



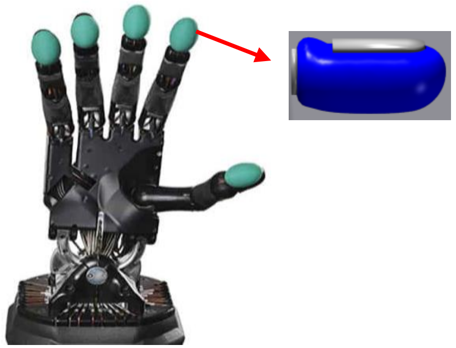
Young's modulus and Poisson ratio, etc.

-> Second stage: grasp fine-tune and tactile sensing with soft body (FEM analysis)

calculate forces, torques of contact points and tactile forces on the gripper.

Sim2real Multimodal Learning for Dexterous Gripper

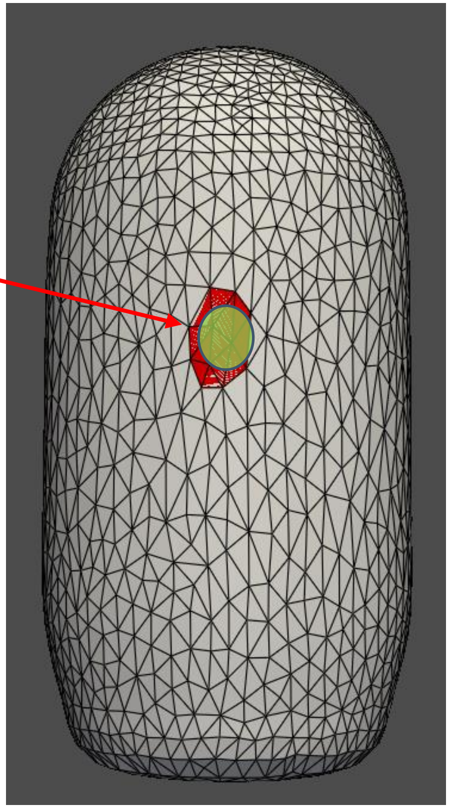
2.1 Two-stage simulation and dataset collection



With biotac tactile sensors
(3D force sensing,
wide sensing regions)

Without biotac tactile sensors
(1D force sensing,
limited sensing regions)

Estimate the 1d force based on the net forces of local region in the tetrahedron FEM model.



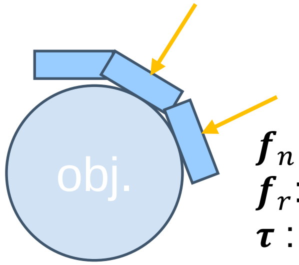
Sim2real Multimodal Learning for Dexterous Gripper

2.1 Two-stage simulation and dataset collection

-> Second stage: grasp fine-tune and tactile sensing with soft body (FEM analysis).

calculate forces, torques of contact points and tactile forces on the gripper.

A finger



$\boldsymbol{f}_n$: normal force
$\boldsymbol{f}_r$: radius force resulting from friction
$\boldsymbol{\tau}$: torque

For a stable grasp under qua-static physics (准静态物理环境):

$$\boldsymbol{F} = \sum \boldsymbol{f}_r^i + \boldsymbol{f}_n^i = \sum \mu_i \boldsymbol{f}_n^i + \boldsymbol{f}_n^i = 0$$

$$\boldsymbol{T} = \sum \boldsymbol{r}_i \times \boldsymbol{\tau}_n^i = \sum \boldsymbol{r}_i \times (\mu_i \boldsymbol{f}_n^i + \boldsymbol{f}_n^i) = 0$$

$-\mu \leq \mu_i \leq \mu$

IPOPT

Optimization

$$min \sum \|\mu_i\|$$

$$q = e^{-\frac{mini\ \sum\|\mu_i\|}{\sum \mu}},\ 0<q<1$$

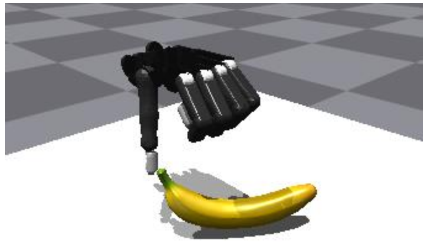
A good grasp should keep force/torque stable with less friction

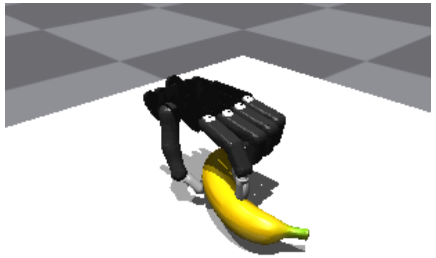
Sim2real Multimodal Learning for Dexterous Gripper

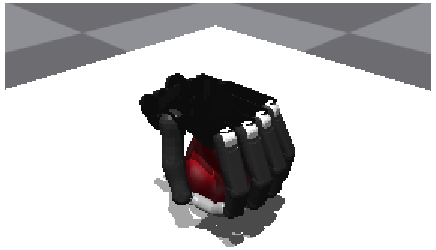
2.1 Two-stage simulation and dataset collection

Some grasp examples

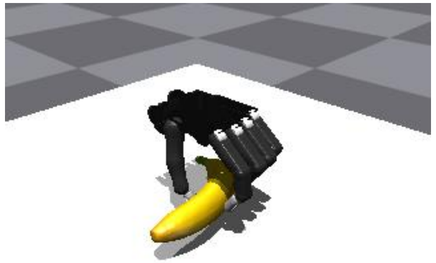
Successful grasp: $q > 0.5$, at least two fingers contact the target object.
Failed grasp: $q < 0.5$, one a finger contacts the target object.

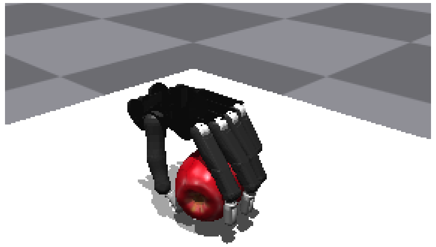

Failed

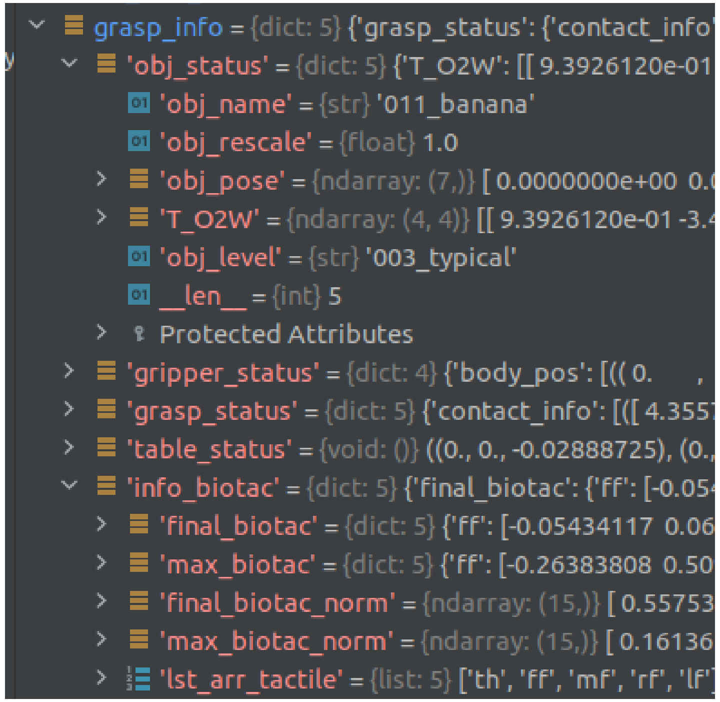

Grasp by 3 fingers


Strong grasp


Grasp by 5 fingers


Grasp by 5 fingers


Finger's side clip the object



Parameters for key frames

Over 500k grasps were simulated with 900G data.

Sim2real Multimodal Learning for Dexterous Gripper

1. Research background

    1.1 State of the art
    1.2 Motivation

2. Method

    2.1 Two-stage simulation and dataset collection
    2.2 Deep learning

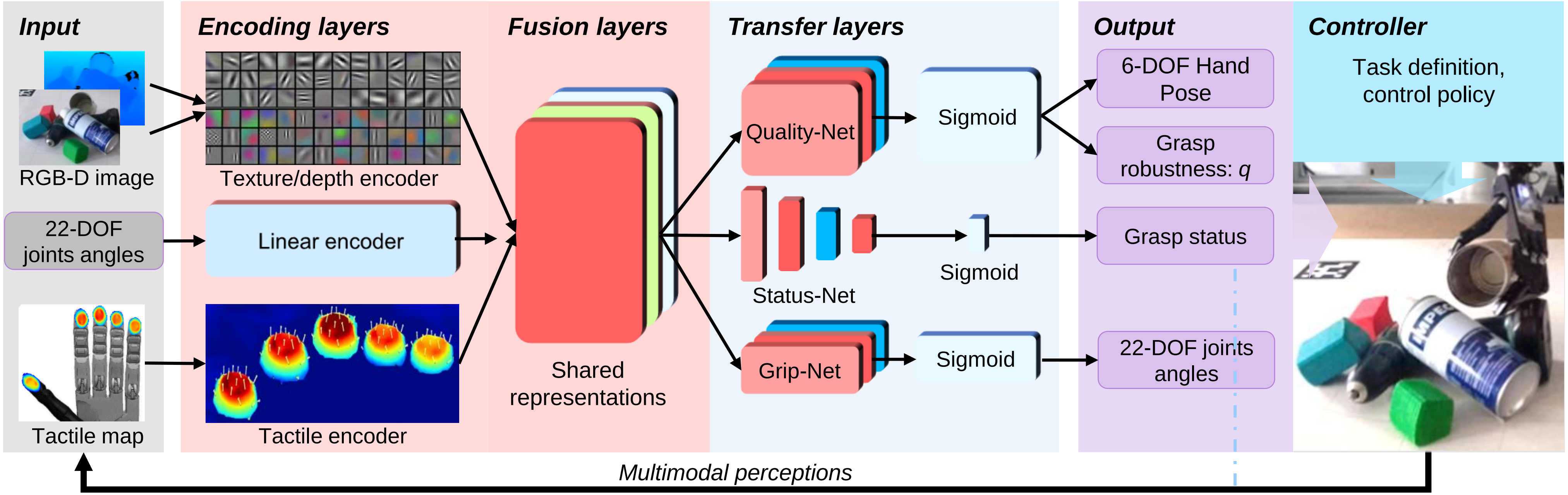
3. Conclusion

    3.1 Simulation
    3.2 NN learning
    3.3 Isaac Gym

Sim2real Multimodal Learning for Dexterous Gripper

2.2 Deep learning

Learn what?

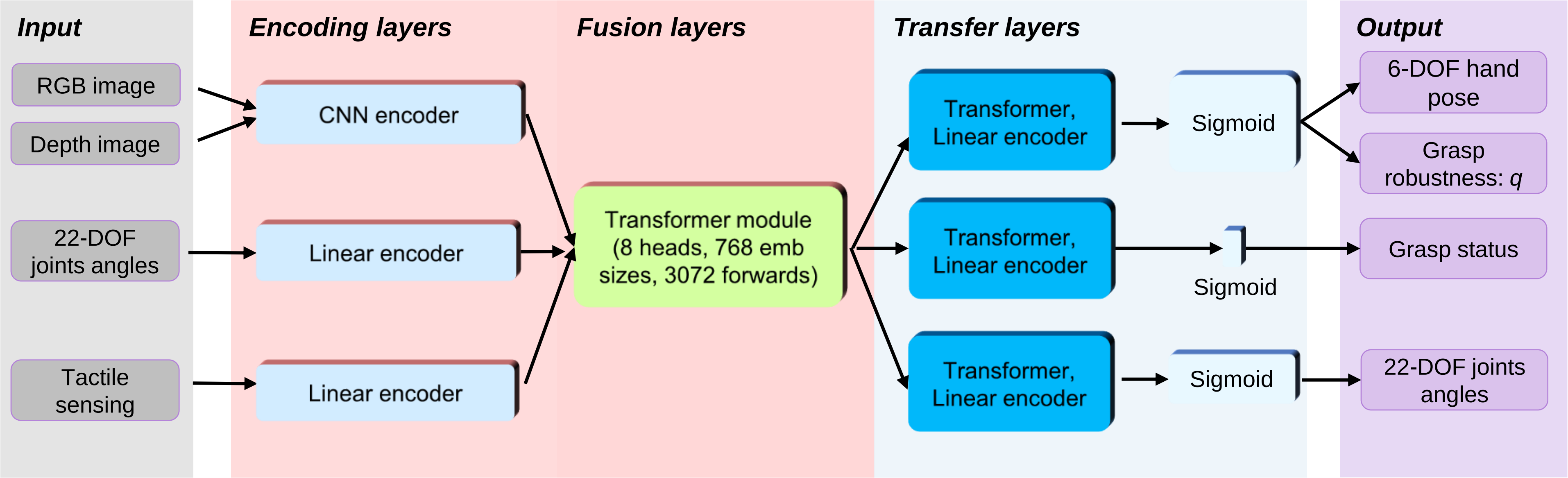


Architecture of the proposed Transformer-based NN for dexterous grasping

Grasp status is to estimate if a grasp will be success before the target object is picked up.

Sim2real Multimodal Learning for Dexterous Gripper

2.2 Deep learning

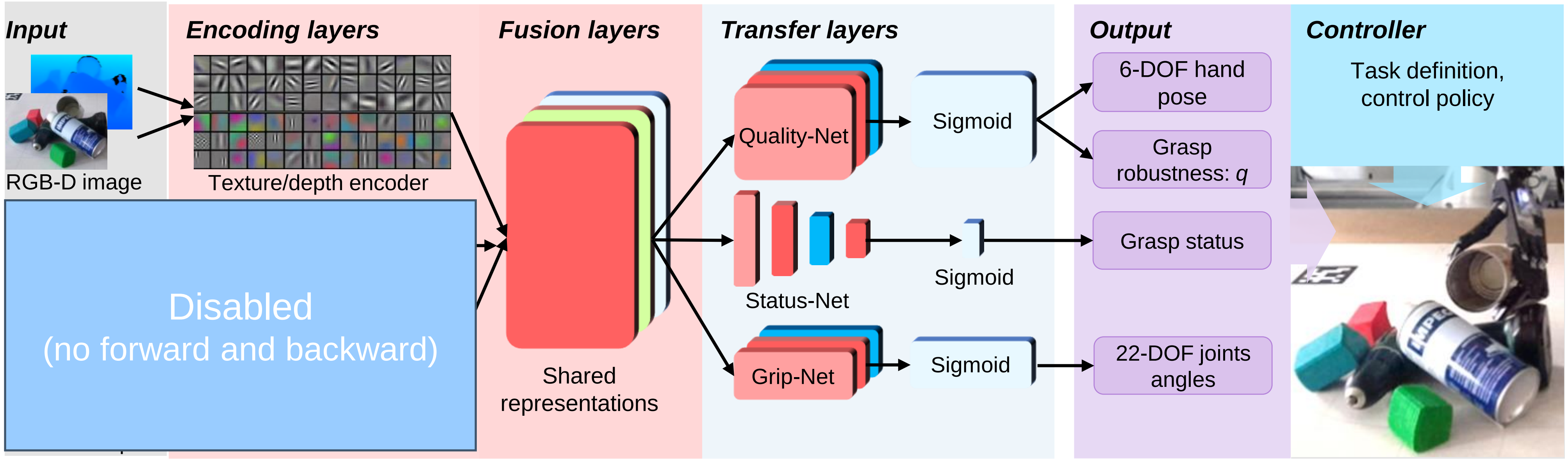


Architecture of the proposed Transformer-based NN for dexterous grasping

Sim2real Multimodal Learning for Dexterous Gripper

2.2 Deep learning (two-stage training)

The first estimation -> before touching objects

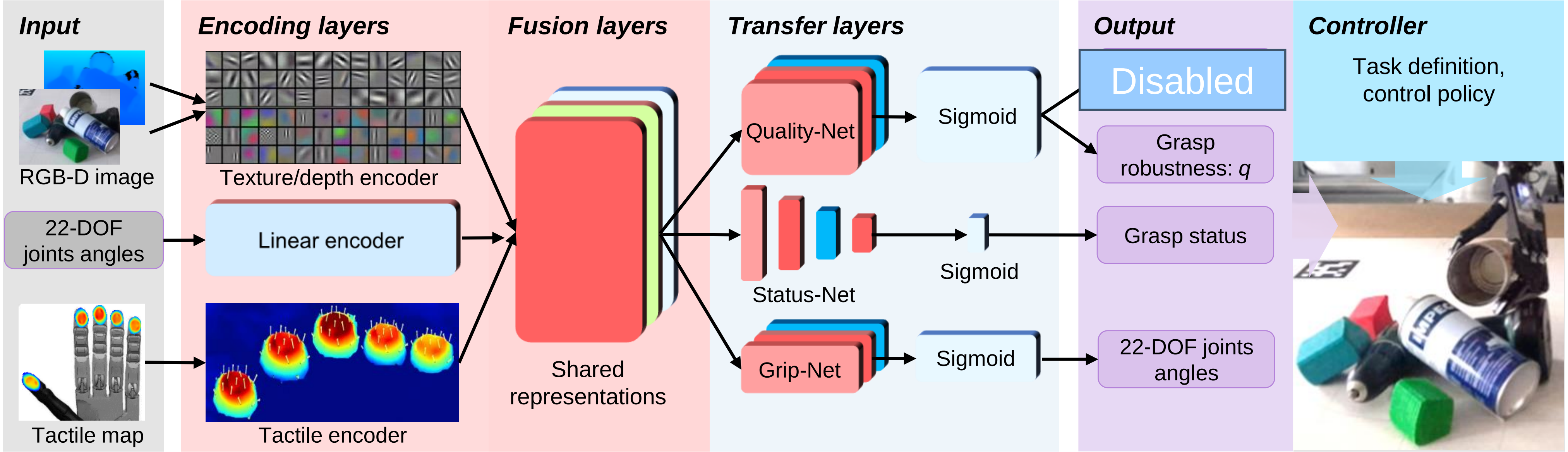


Total Loss=Loss_6d_pose + Loss_q + Loss_status + Loss_22d

Sim2real Multimodal Learning for Dexterous Gripper

2.2 Deep learning (two-stage training)

The second estimation -> when touching an object



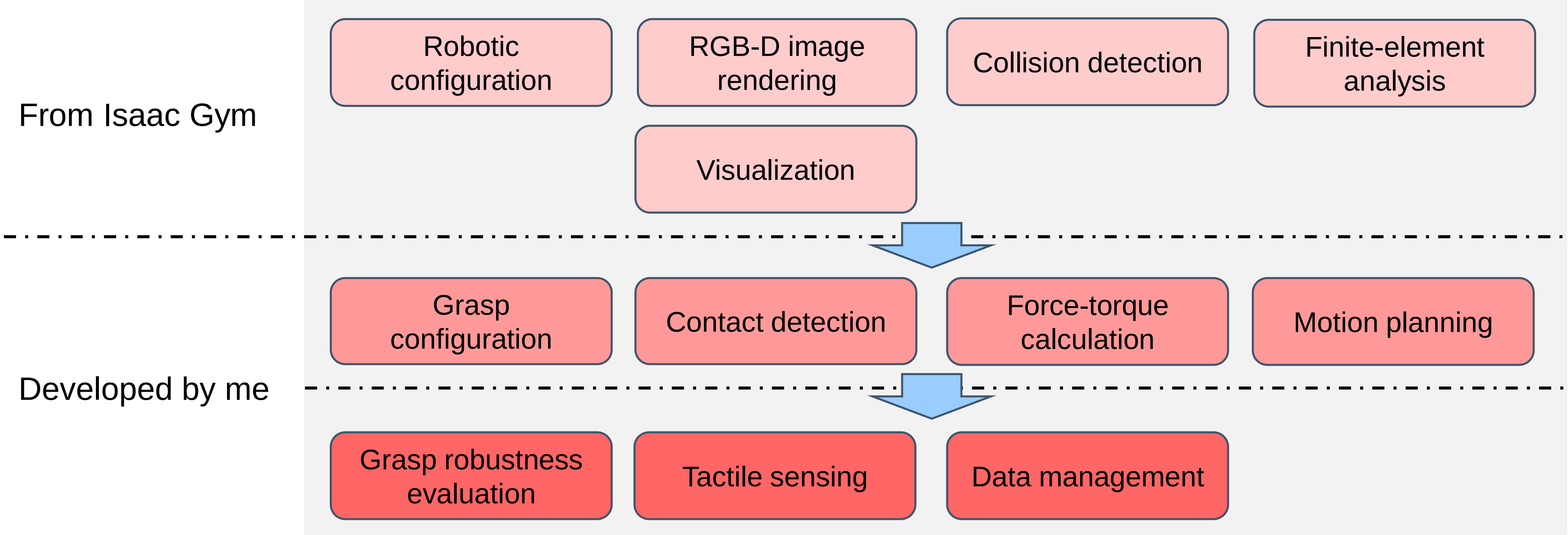
$$Total\ Loss = Loss\_q + Loss\_status + Loss\_22d$$

Sim2real Multimodal Learning for Dexterous Gripper

Sim2real Multimodal Learning for Dexterous Gripper

3. Conclusion

3.1 Simulation



Pluggable simulation framework with a two-stage simulation:
-> Fast contact, soft tactile sensing.
-> Practical DOF definition and motion plan.

Sim2real Multimodal Learning for Dexterous Gripper

Sim2real Multimodal Learning for Dexterous Gripper

3. Conclusion

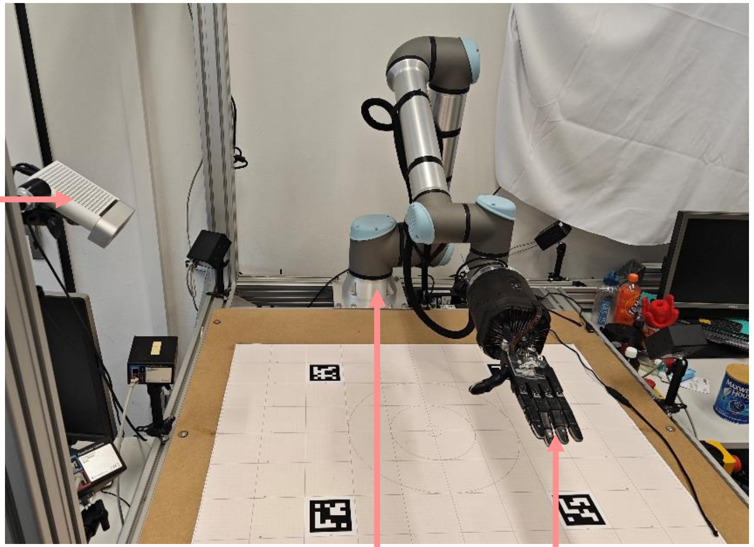
3.2 NN learning (trained on 10k datasets)

Error	Frist estimation (RGB-D)	Second estimation (RGB-D+ Joints' angles + Tactile)
6-DOF hand pose (rad)	0.18	0.18
Grasp robustness *q*	0.07	0.05
Grasp status (%)	0.81	0.89
22-DOF joints (rad)	0.43	0.44

Sim2real Multimodal Learning for Dexterous Gripper

3. Conclusion

3.2 NN learning (more training data -> 500k, experiments with real-world robots, etc.)



Grasp experiments with complex scenarios：success rate, predictions erros?

Sim2real Multimodal Learning for Dexterous Gripper

1. Research background

   1.1 State of the art
   1.2 Motivation

2. Method

   2.1 Two-stage simulation and dataset collection
   2.2 Deep learning

3. Conclusion

   3.1 Simulation
   3.2 NN learning
   3.3 Isaac Gym

Sim2real Multimodal Learning for Dexterous Gripper

3. Conclusion

3.3 Isaac Gym (About the lifetime of a variable)

1) Typical python script

```
Class A()
    def __init__():
        other codes…


def main ():
    for i in range(XX):
        a = A()    # define a variable
        …
        del A      # release memory
```

2) Typical use case for Reinforcement Learning with Isaac Gym

```
Class A()
    def __init__():
        some variable/functions/classes from Isaac Gym
        …

def main ():
    a = A()        # define a variable
    while loop:
        reinforcement learning progress
        …
    # program end
```

Sim2real Multimodal Learning for Dexterous Gripper

3. Conclusion

3.3 Isaac Gym (About the lifetime of a variable)

3) My use case for Reinforcement learning with Isaac Gym

```
Class A()
    def __init__():
        some variable/functions/classes from Isaac Gym
        …

def main ():
    for i in range(XX):
        a = A()    # define a variable
        …
        del A      # memory is not fully released!
```

Sim2real Multimodal Learning for Dexterous Gripper

3. Conclusion

3.3 Isaac Gym (About the lifetime of a variable)

4) Improved version

```
Class A()
    def __init__():
        some variable/functions/classes from Isaac Gym
        …

def main ():
    a = A()        # define a variable
    …
    # program end
```

Use a shell script to execute the python script containing the main function. When a python scripted ends in the shell script loop, the lifetime of Isaac Gym variables will be stopped.

Thanks for your attention