ROMANDIC winter school 2026

Kranjska Gora, Slovenia, Feb. 10th 2026

Learning of unfolding, flinging and flattening of textiles as part of complete textile handling pipeline

Andrej Gams,

Institut "Jožef Stefan"

andrej.gams@ijs.si

https://andrejgams.ijs.si,

https://hcr.ijs.si, https://abr.ijs.si

Institut "Jožef Stefan"
Ljubljana, Slovenija

ROMANDIC

Robot Manipulation of Deformables through Dynamic Actions

Outline

Textile handling pipeline

- Picking up and unfolding in the air
  as continuation of work for the ICRA 2024 Robotic Grasping and Manipulation Competition

- Dynamically spreading a cloth flat on the table with flinging

- Autonomous flattening

ROMANDIC

Handling of textiles

- We basically couldn't do this before

- It looks great and people want it!

- It avoids what robots are still bad at

Robot behaviors are produced by models trained via imitation learning... and they often need many, many trajectories.
The work ALOHA Unleashed by Google is a great example, **needing about 6,000 demonstrations** to learn how to, for example, tie a pair of shoelaces.

GUEST ARTICLE   ROBOTICS

Why Is Everyone's Robot Folding Clothes? ›And what does it tell us about the state of modern robot learning?

BY CHRIS PAXTON | 19 NOV 2025 | MIN READ
Chris Paxton is currently leading AI innovation at Agility Robotics.

There are good reasons why Figure's humanoid can fold towels this well. FIGURE

ROMANDIC

3

Examples

Weave robotics

Pi-0

autonomous, 2x speed

Figure 02

Figure 03

ROMANDIC

Research projects on handling of deformables at JSI:

- Robot Textile and Fabric Inspection and Manipulation (**RTFM**) – *ARIS (Slovenian national funding) project* (2022 – 2026) (with FRI-UL)

- Flexible robotic solutions for the recycling of soft materials – **FlexCycle** (2025 – 2029) (*just started*)

- RObot MANipulation of Deformables through dynamIC actions – **ROMANDIC** (2024 -2027)

ROMANDIC Research at JSI

- Autonomous flattening of garments

- Unfolding of garments

- Dynamic Manipulation
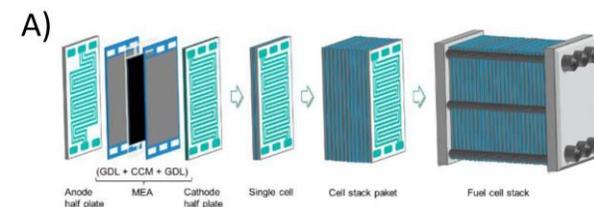
Deformable object manipulation for textiles

Textiles: Clothing, garments, towels, sheets, etc.

- One of the most common and challenging DOs

- Highly complex deformations - effectively infinite degrees of freedom / variations in deformation

- Wrinkling, folding, stretching, self-occluding, drape, …

- Deformation also dependent on material properties (stiffness, friction, …)

- Applications in
  - Healthcare: (e.g. robot assistive dressing)
  - Textile recycling
  - Service industry (e.g. laundry folding)
  - Industrial robotics (assembly, fashion)

Picking up and unfolding in the air

as continuation of work for the ICRA 2024 Robotic Grasping and Manipulation Competition
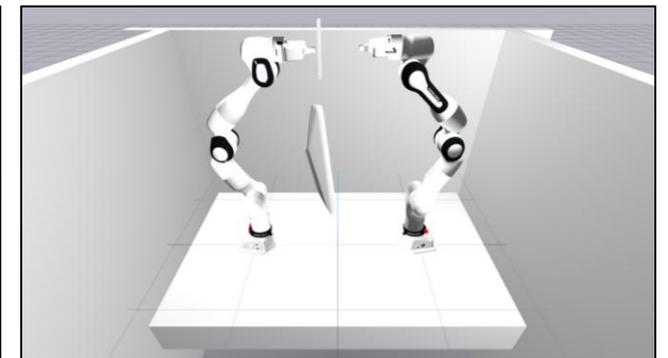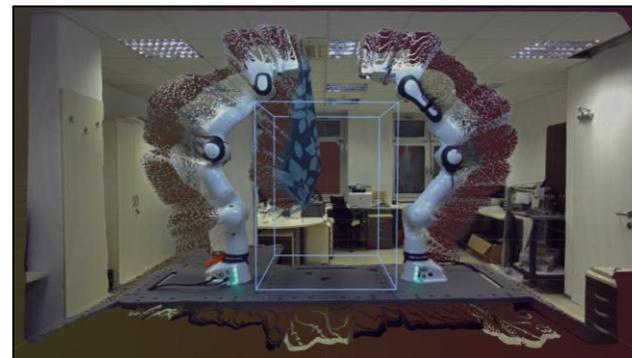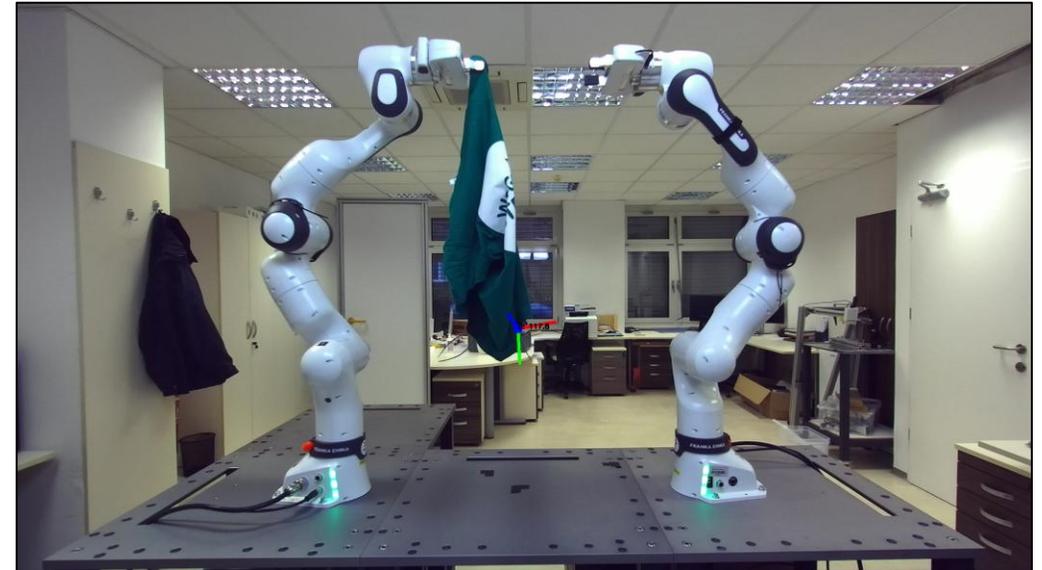
ROMANDIC

Unfolding garments in the air

- How to grasp textile/garment so that it is most unfolded when extending it

- Bimanual manipulation

- Goal: dynamic handling of textiles

Robotics challenges

- Redundancy of the bimanual system

- Inverse kinematics

- Joint limits

- Collision avoidance (robots, environment, textile)

ROMANDIC

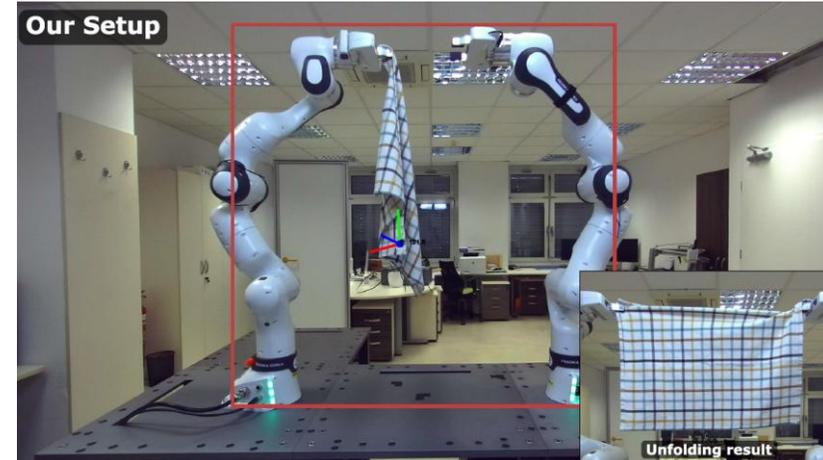# ICRA 2024 Cloth Unfolding Benchmark

Overview of the unfolding procedure
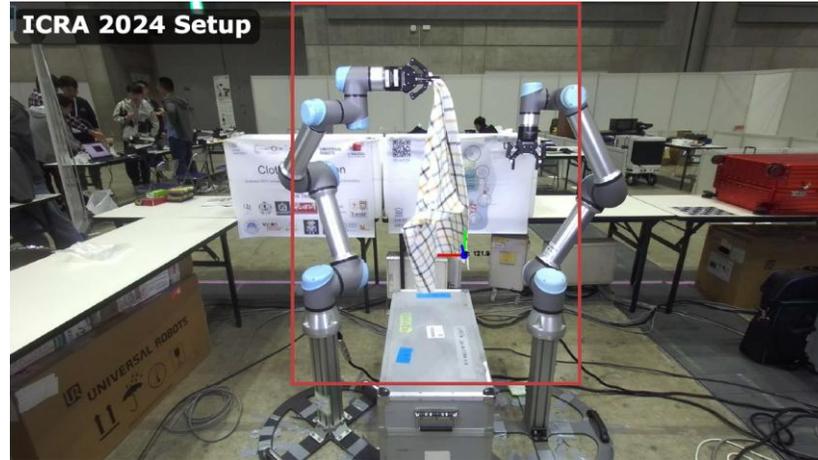
1. Grasp high ... lt observation
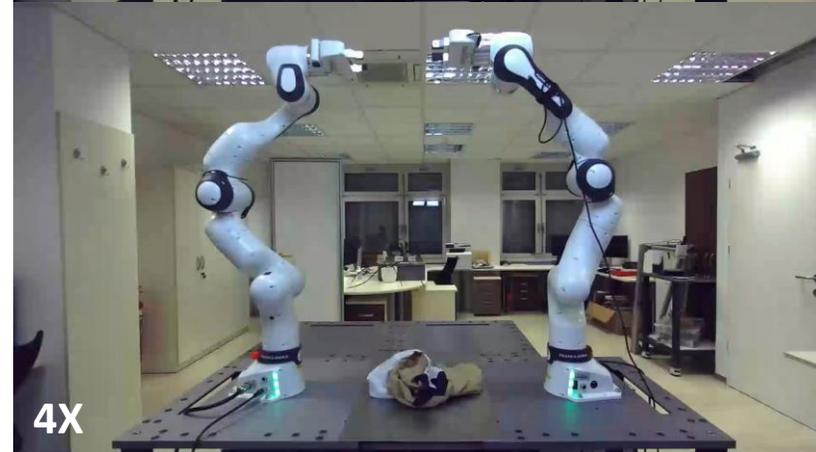
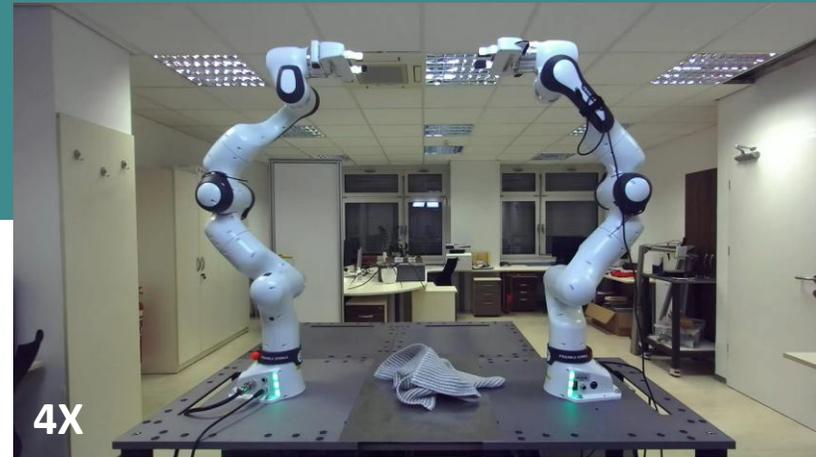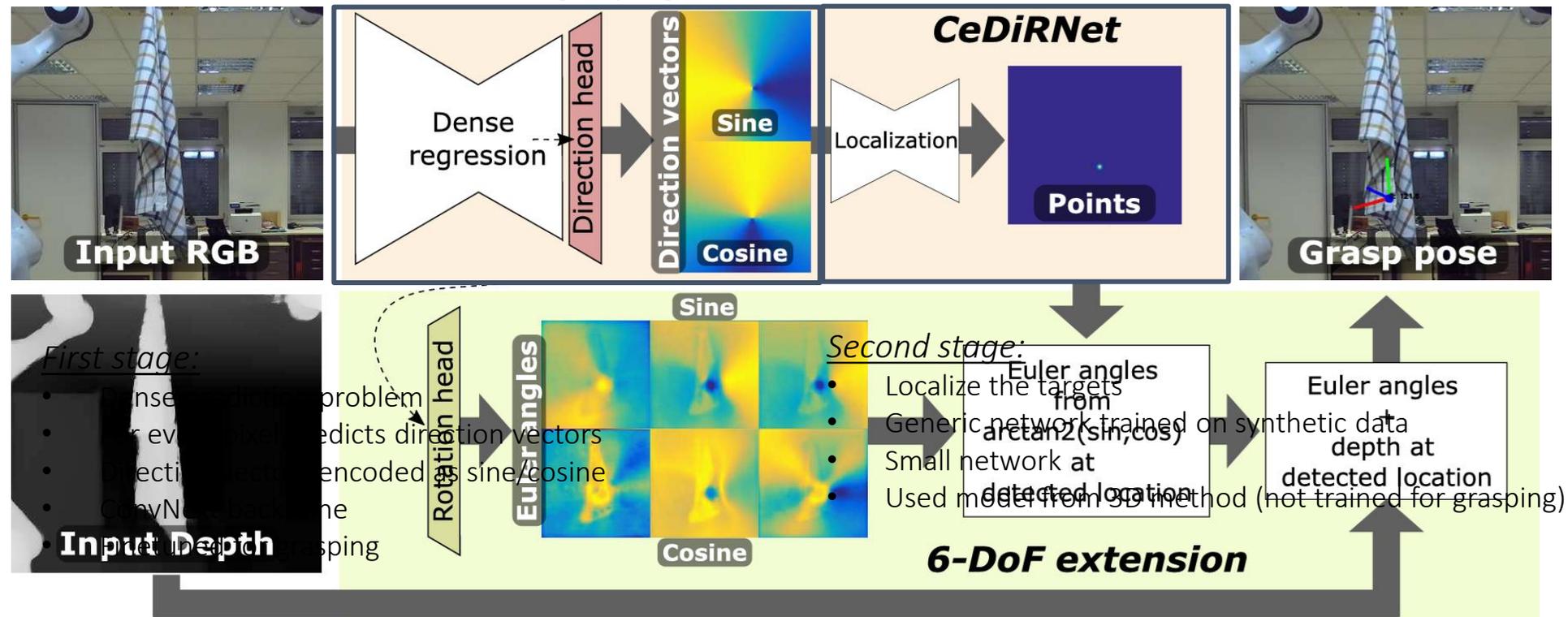Successful unfolding (by team Ljubljana)

ROMANDIC

Re-implementation

Unfoding in the air

- Initially grab first the highest and the the lowest points of the garment and let it swing-out.

- Predict the grasp point(s) (next slide).

- Select a grasp pose with the highest confidence score from CeDiRNet-6DoF.

- Converted into the robot's frame.

- Analytical inverse kinematics solver applied to map the target 6-DoF pose into the corresponding joint space configuration (redundant system – closest solution to predefined example).

- RRT-connect to plan the motion;
  - real-time collision checking is performed against a model of the environment—including both robot arms, table and detected cloth.

- Fully extend and measure coverage.

ROMANDIC

Selection of grasp point – CeDiRNet-6DoF

- Extension of CeDiRNet [1] with a 6-DoF grasping model.

First stage:
- Dense prediction problem
- For every pixel predicts direction vectors
- Direction vectors encoded as sine/cosine
- ConvNet backbone
- Not trained for grasping

Second stage:
- Localize the targets
- Generic network trained on synthetic data
- Small network at detected location
- Used model from 3D method (not trained for grasping)

Added Euler angles for 6-DoF: encoded as trigonometric function (sin/cos) [2]

- Training only around groundtruth, during inference predicting
- Network based on RGB/RGB-D to predict 5-DoF, using depth from input for final 6-DoF

Collaboration with FRI-UL

[1] D. Tabernik, J. Muhovič, and D. Skočaj, Dense center-direction regression for object counting and localization with point supervision, Pattern Recognition, 2024.
[2] Tabernik, Nimac, Jerićević, Skočaj, Gams; Vision-Based 6-DoF Grasp Pose Estimation for Robot Cloth Unfolding, submitted

Robust solution

Demonstration at IFAM 2025 fair in Ljubljana

Results

ICRA 2024 Set

cloth-checkers-airo | cloth-checkered-rag | cloth-waffle-rag | cloth-green-airo | tshirt-blue-airo | tshirt-zara-airo | tshirt-green-airo | tshirt-women-airo

Our Set

cloth-rainbow-checkered | cloth-rainbow-striped | cloth-checkered-green | shirt-pink | tshirt-beige | tshirt-black-large | tshirt-black-small | tshirt-red

Results
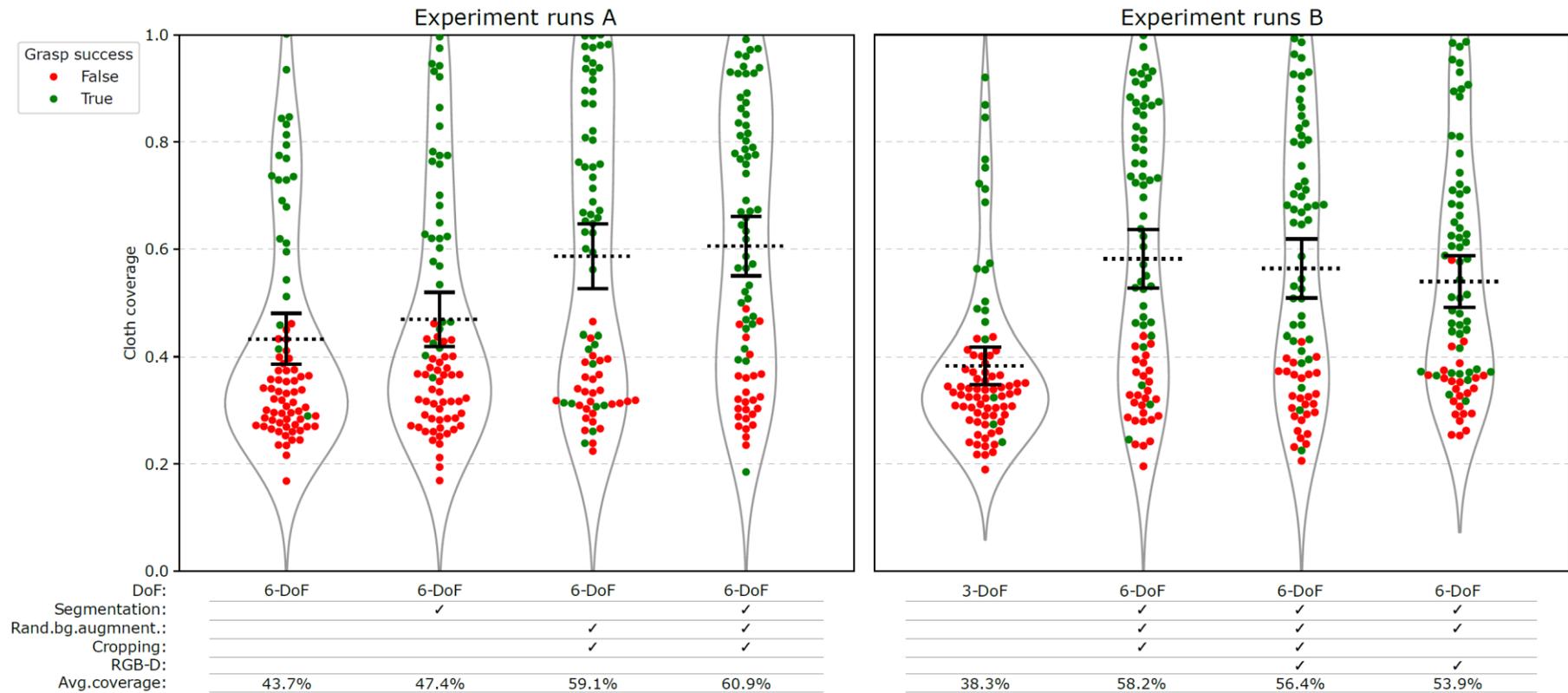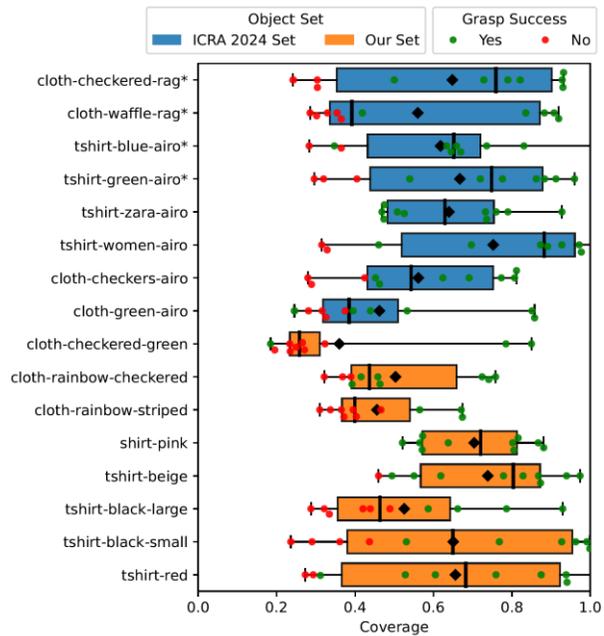
Rank	Team Name/Model	Coverage ICRA 2024 setup	Coverage Our setup
1th	AIR-JNU	**0.60**	-
2th	**CeDiRNet-6DoF (our)**	0.57	$0.61 \pm 0.05_{95\% CI}$
3th	Ewha Glab	0.55	-
4th	SCUT-ROBOT	0.53	-
5th	Team Greater Bay	0.53	-
6th	Samsung Research China	0.48	-
7th	Shibata Lab	0.46	-
8th	AI&ROBOT LAB	0.45	-
9th	UOS-Robotics	0.39	-
10th	AIS Shinshu	0.37	-
11th	3C1S	0.35	-

	DoF	Seg.	Random bg. augment.	Crop. by robot pos.	RGB-D	Coverage ICRA 2024	Coverage Our	Coverage Both	Grasp Success ICRA 2024	Grasp Success Our	Grasp Success Both	Coverage with Succ. Grasp ICRA 2024	Coverage with Succ. Grasp Our	Coverage with Succ. Grasp Both
RUNS A	6-DoF					0.455	0.420	0.437	0.388	0.225	0.306	0.678	**0.735**	0.709
	6-DoF	✓				0.490	0.458	0.474	0.475	0.325	0.400	0.660	0.646	0.653
	6-DoF		✓	✓		0.604	0.578	0.591	0.675	**0.625**	0.650	0.698	0.706	0.702
	6-DoF	✓	✓	✓		**0.635**	**0.583**	**0.609**	**0.750**	**0.625**	**0.688**	**0.750**	0.683	**0.717**
RUNS B	3-DoF					0.388	0.379	0.383	0.325	0.150	0.238	0.553	0.646	0.581
	6-DoF	✓	✓	✓		**0.594**	**0.570**	**0.582**	0.700	0.625	0.663	0.708	**0.726**	**0.717**
	6-DoF	✓	✓	✓	✓	0.593	0.534	0.564	0.675	0.650	0.663	**0.717**	0.612	0.661
	6-DoF	✓	✓		✓	0.577	0.502	0.539	**0.725**	**0.675**	**0.700**	**0.717**	0.569	0.643

Dynamically spreading a cloth flat on the table

ROMANDIC

Motivation

- Spreading is a common, often essential part of the tectile handling pipeline.

- Difficult to automate (high DoF dynamics) → **need for adaptive, coordinated, fast motion.**

- Additional complexity increase in **bimanual settings**

- Prior work often **quasi-static** or relies on **hand-crafted primitives**

- **Dynamic**, high-speed strategies promising for spreading/placement of textiles.

Isaac Sim/Isaac Lab & reinforcement learning

- **Task Objective**: Two robots must dynamically spread a cloth flat on the ground

- **Key challenge**: nonlinear cloth dynamics, coupling between arms, need for precise timing & coordination.

  - Traditional motion planning would be complex - **learning-based control**.

- We employ **Isaac Sim** and **Isaac Lab** for environment setup and reinforcement learning (RL)

  - Isaac Sim – dynamics simulator can use various robots
  - Isaac Lab – wrapper for Isaac SIM used for DRL

Proposed methodology

- **Task:** dynamic bimanual cloth placement—two arms **learn** fast, fluid fling-and-place motions onto a surface.

- **Approach:** PPO policy outputs Dynamic Movement Primitive (DMP) parameters
  - smooth, time-scalable trajectories in a compact action space.

- **Reward design:** composite metrics evaluated sparsely at episode end to stabilize learning.

- Placement motion needs precise timing and directionality—hard to script; RL+DMPs can obtain hard-to-handcraft trajectories.

- Parallel simulation cuts sample cost.
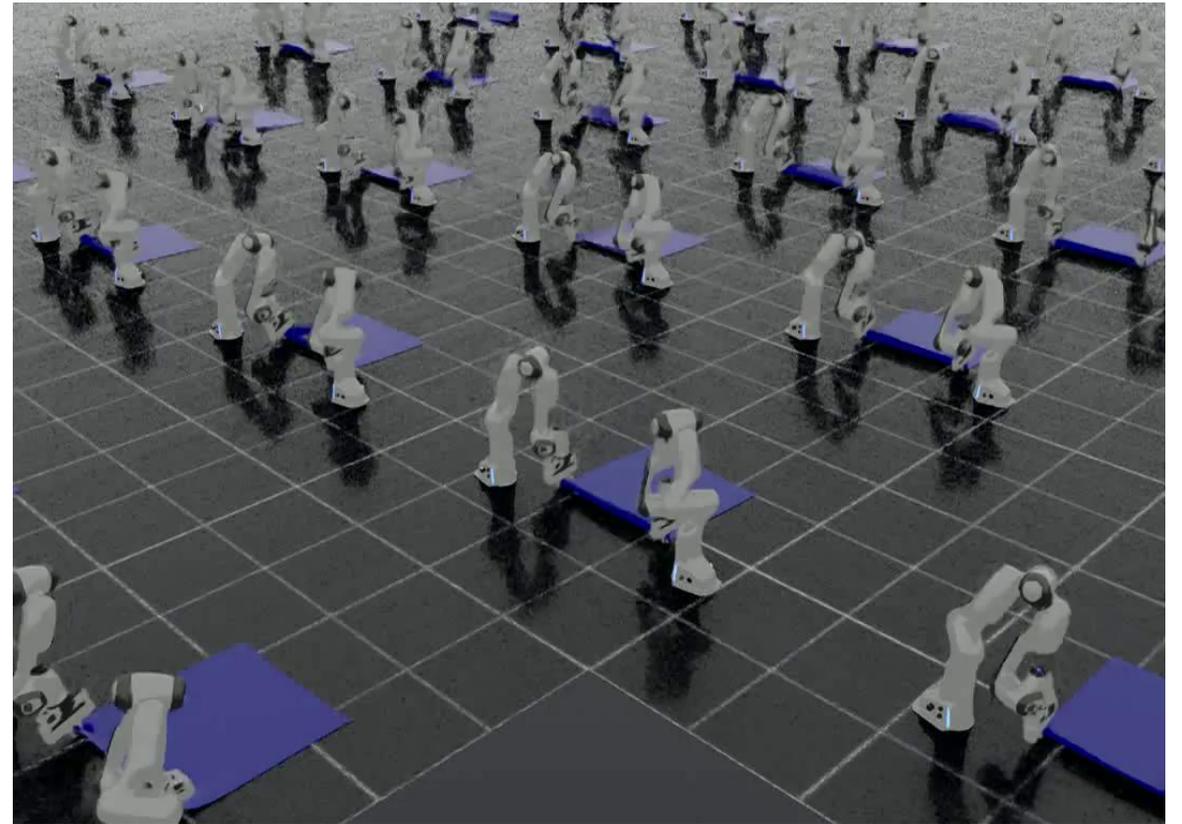
Robot control and reward calculation

- **Separate arm vs. bimanual control**
    1. Control each arm separately using calculated actions
    2. Define absolute and relative trajectories to ensure arms move symmetrically

- **How often to calculate actions/rewards**
    1. Actions (joint targets) and rewards calculated in **each simulation step**
    2. Calculate actions only **once per episode** (movement = DMP parameters)
        1. Integrate DMP each simulation step to determined target joints
        2. Calculate rewards at the **end of the episode**

These choices reduce complexity of training (14 DoF -> 2DoF) and complexity of robot control

ROMANDIC

Absolute and relative coordinates

- Simplification of dual-arm robot motion description
  - Absolute trajectories describe motion of midpoint between end effectors
  - Relative trajectories describe relations between end effectors

Nemec, Likar, Gams, Ude. "Adaptive human robot cooperation scheme for bimanual robots." *Advances in Robot Kinematics* 2016. Cham: Springer International Publishing, 2017. 371–380.

$$\mathbf{y}_a = \frac{\mathbf{y}_1 + \mathbf{y}_2}{2}, \quad \mathbf{R}_a = \mathbf{R}_1 \mathbf{R}^1_{\mathbf{k}^1_{12}}(\theta_{12}/2)$$

$$\mathbf{y}_r = \mathbf{y}_2 - \mathbf{y}_1, \mathbf{R}_r = \mathbf{R}^2_1$$

$$\mathbf{J}_a = [\tfrac{1}{2}\mathbf{J}_1 \quad \tfrac{1}{2}\mathbf{J}_2], \quad \mathbf{J}_r = [-\mathbf{J}_1 \cdot \mathbf{J}_2], \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_a \\ \mathbf{J}_r \end{bmatrix}$$

- Absolute and relative Jacobian matrices are used for inverse kinematics to obtain joint angles for both arms simultaneously
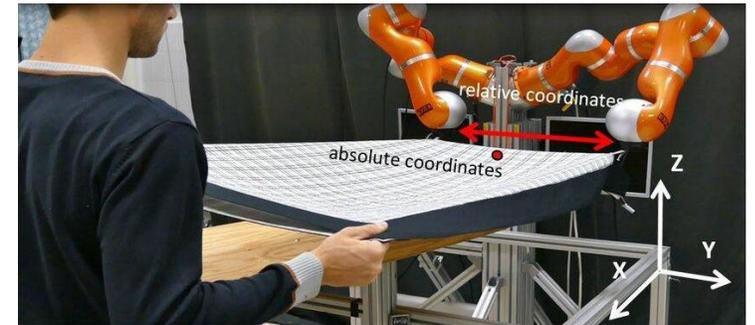
Training setup (1/2)

- We wish to generalize to different cloth lengths

- How to assert information about cloth dynamics without monitoring its motions?
Probing phase: brief pre-motion excitation to infer cloth properties
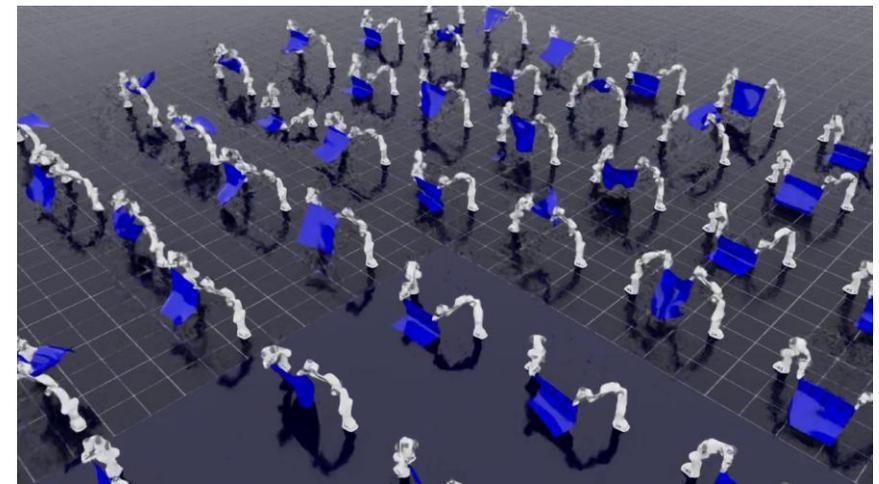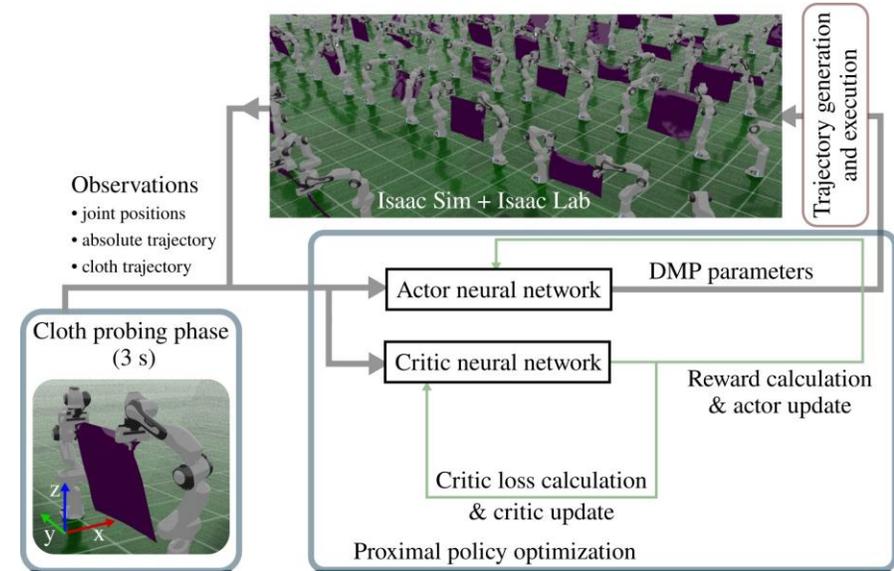
- Motions are defined using **Dynamic Movement Primitive (DMP) parameters**
    - The policy outputs DMP parameters which are integrated to obtain an absolute Cartesian trajectory
    - Robots are simultaneously controlled – control of absolute motion while maintaing relative postures fixed

- Training is split into two parts:
    - Probing phase (0-3 s) – the robots test the cloth's dynamics by observing corner motions during a predefined absolute motion
    - Placement phase (3 s -) – training in 3-second episodes
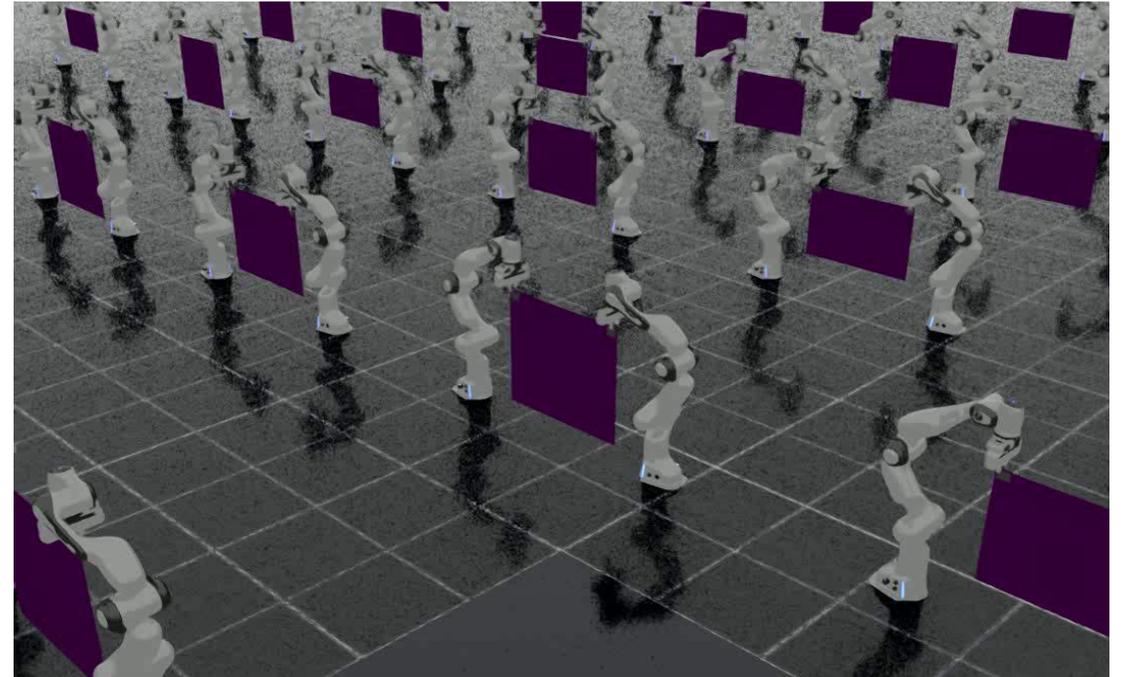
RL & Proximal Policy Optimization (PPO)

- Key components in RL:
  - **Agent**: Learner or decision maker
  - **Environment**: The system the agent interacts with
  - **Action**: What the agent does in the environment
  - **Reward**: Feedback given after actions
  - **Policy**: Strategy the agent uses to make decisions
- PPO – popular actor-critic RL algorithm, improves training stability by using a clipped objective function
- Directly updates the policy based on reward gradients.
- Balanced exploration and exploitation
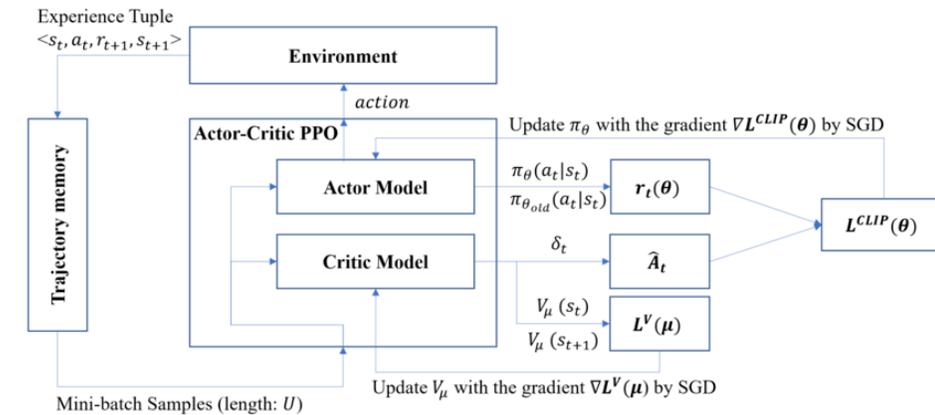
Expected reward

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$$

Objective

$$L^{\text{clip}}(\boldsymbol{\theta}) = \mathbb{E}_t\left[\min\left(\rho_t(\boldsymbol{\theta})\hat{A}_t, \text{clip}(\rho_t(\boldsymbol{\theta}), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right]$$

$$\hat{A}_t = \sum_{l=0}^{T-1}(\gamma\lambda)^l \delta_{t+l}$$

$$\delta_t = r_t + \gamma V(\mathbf{s}_{t+1}) - V(\mathbf{s}_t)$$

Actions and observations

- Observations:
  - Cartesian poses of the absolute robot frame, gathered during the 3-second probing phase, sampled at 2.5 Hz
  (X, Z only, 2 axes × 8 time steps = 16);
  - Positions of the two cloth's free corners positions, also gathered during the 3-second probing phase, sampled at 2.5 Hz
  (2 corners × 3 axes × 8 time steps = 48).

- Actions:
  - 55 DMP parameters consisting of initial and goal absolute position (X-Z plane), duration and DMP weights

$$\mathbf{a} = \left[\mathbf{y}_0, \, \mathbf{g}, \, \{\mathbf{w}_i\}_{i=1}^{25}, \, \tau \right] \in \mathbb{R}^{55}$$

Environment & Rewards

- **Physics & Setup**:
  - Cloth is defined as a deformable object, length is randomized (0.45 m – 0.7 m)
  - Robots are facing each other, moving the cloth between them using motions in X-Z plane

- **Reward Components**:
  - **Area** reward: Maximizes pairwise distances between the 4 cloth corners.
  - **Corner X**-axis reward: Encourages outward X movement of free corners for better flattening.
  - **Height** reward: Penalizes lifted cloth regions to promote flat placement.
  - **Direction** reward: Gives reward when free corners are far from the robot.
  - **Action** penalty: Limits excessive joint velocity (energy-efficient behavior).
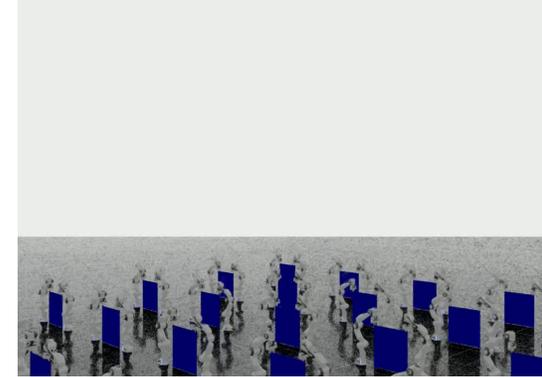
$$r_{\text{area}} = \sum_{i=0}^{3} \sum_{j=i+1}^{3} \|\mathbf{a}_i - \mathbf{a}_j\|_2$$

$$r_{\text{height}} = \frac{1}{1+\overline{h}}$$

$$r_{\text{corner-x}} = \left(\frac{1}{2} \sum_{i=2}^{3} \left(\mathbf{a}_{x,i} - x_{\text{origin}} \right) \right)$$

$$r_{\text{direction}} = \left(\frac{1}{2} \sum_{i=2}^{3} \left(\mathbf{a}_{x,i} - x_{\text{origin}} \right) - \frac{1}{2} \sum_{i=0}^{1} \left(\mathbf{a}_{x,i} - x_{\text{origin}} \right) \right)$$

$$r_T = \gamma_a\, r_{\text{area}} + \gamma_h\, r_{\text{height}} + \gamma_x\, r_{\text{corner-x}} + \gamma_d\, r_{\text{direction}}$$

Workflow

- Simulation observations (robot/corner positions)
- Policy neural network
- Obtain DMP parameters for absolute trajectory
- DMP integration, conversion to left/right robot arm trajectories
- Inverse kinematics for joint position calculation & apply robot actions
- Calculate rewards and perform training step (update policy network)
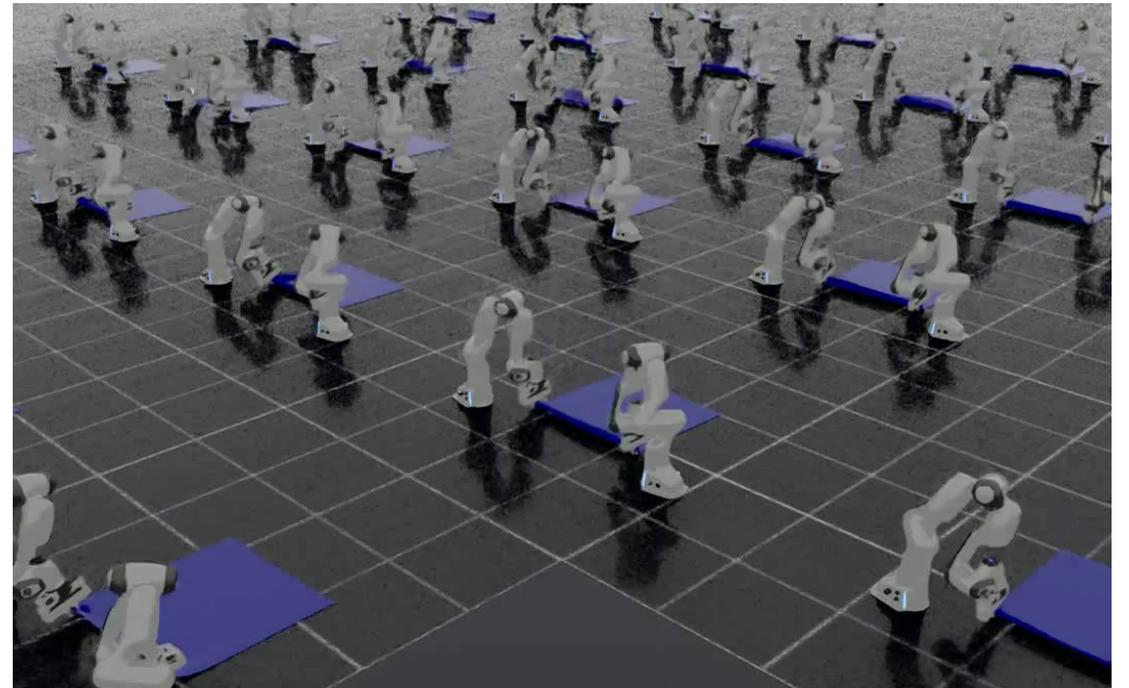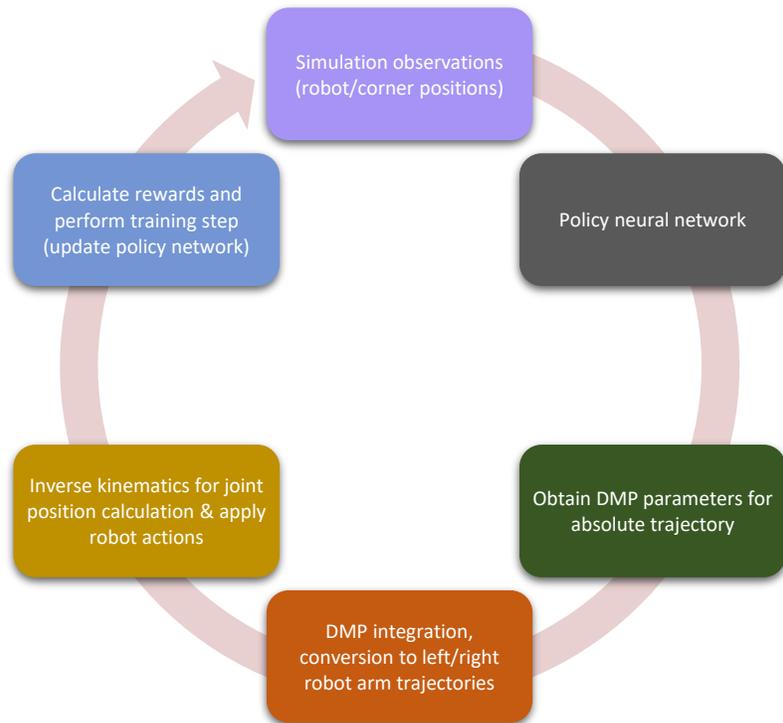
Training regimes

- We compare our training regime against two baselines:
  - Static-start: Identical setup, but no probing is performed. All observations are set to zero and are therefore constant time-wise.
  - Single-length: All environments use the same 0.7 × 0.7 m square cloth. During training, the policy can therefore not generalize to different lengths, meaning that it may perform worse at test time
- These baselines isolate the effect of the probing phase and of length randomization on final placement performance on unseen cloths.
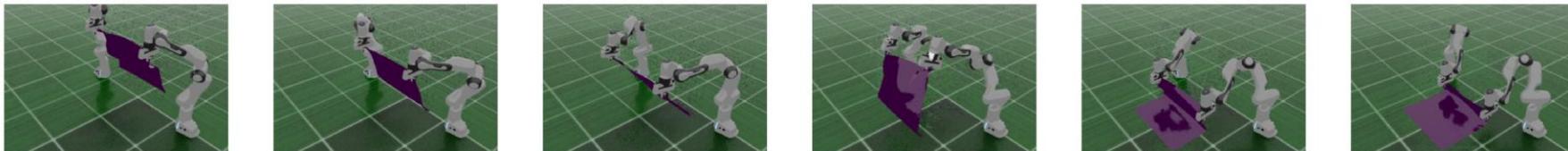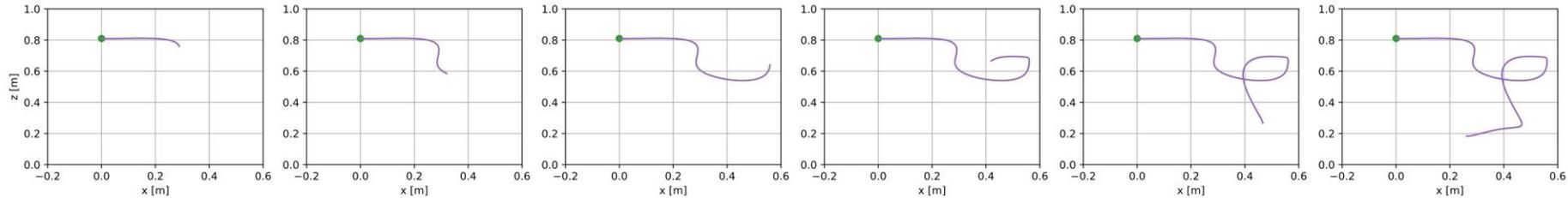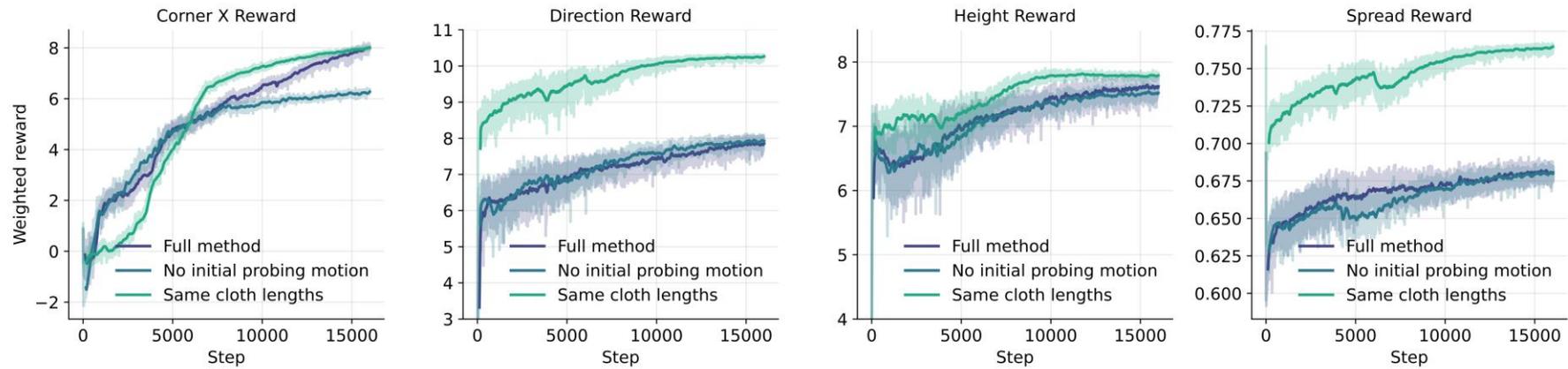
Results (2/2)

TABLE I

REWARDS WITH **UNSEEN** CLOTH LENGTHS (0.45–0.7 M). MEAN ± STANDARD DEVIATION OVER 500 ENVIRONMENTS.

Method	r_{area}	r_{height}	$r_{corner-x}$	$r_{direction}$
Static-start	0.68 ± 0.04	7.49 ± 0.11	6.50 ± 0.71	7.95 ± 1.12
Single-length	0.58 ± 0.1	6.34 ± 1.39	6.82 ± 1.45	3.75 ± 3.53
Full	$\mathbf{0.69 \pm 0.04}$	$\mathbf{7.69 \pm 0.14}$	$\mathbf{8.2 \pm 0.59}$	$\mathbf{8.05 \pm 1.13}$

TABLE II

COVERAGE ON **UNSEEN** CLOTH LENGTHS (0.45–0.7 M). MEAN ± STANDARD DEVIATION OVER 500 ENVIRONMENTS.

Method	Coverage [%]	% of envs. with coverage $> 80\%$
Static-start	92.89 ± 3.17	99.41
Single-length	65.5 ± 19.7	27.54
Full	$\mathbf{94.77 \pm 2.18}$	**99.61**

30
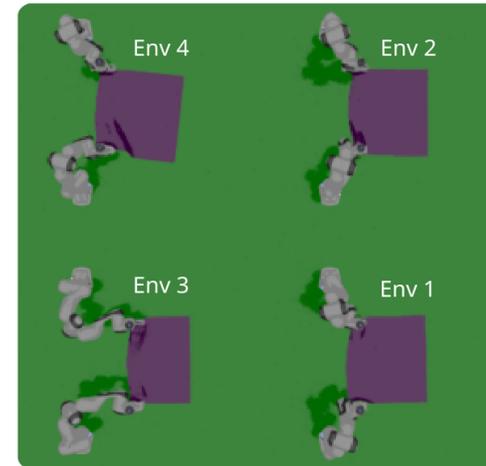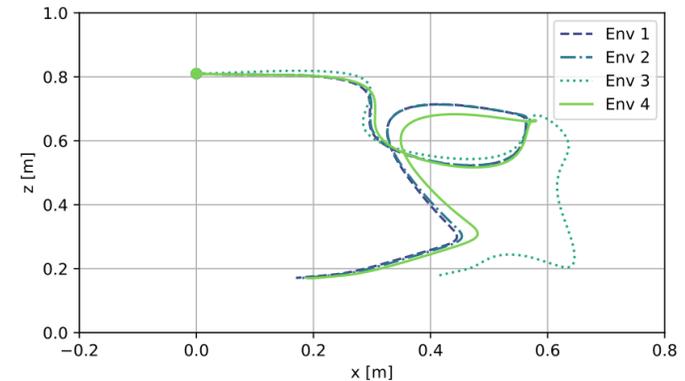
Discussion

- Generalization on unseen cloths (n=500):
  - Full (length randomization + probing) achieves best coverage compared to baselines
- Reward-based comparison on unseen cloths:
  - Full outperforms on all components, notably Corner-X.
- Learning curves insight:
  - Single-length shows the highest training returns (easier task) but fails to generalize to new lengths; training reward ≠ test performance. Full improves Corner-X during training.
- Qualitative trajectories:
  - Policy produces fast, smooth fling-and-place; cloth inverts mid-air and lands flat with minimal wrinkles; trajectories adapt to cloth length (distinct paths for shorter cloths).
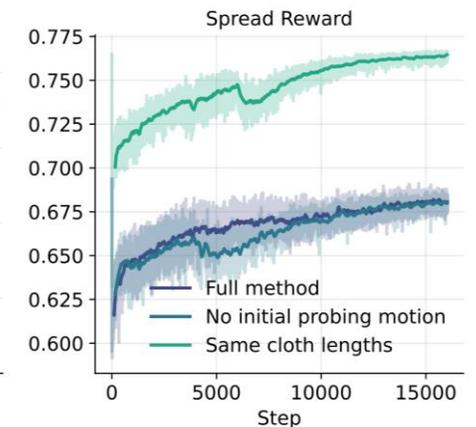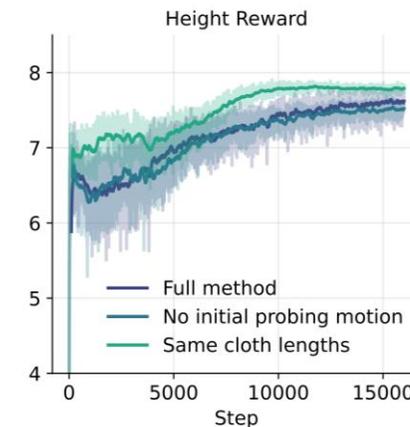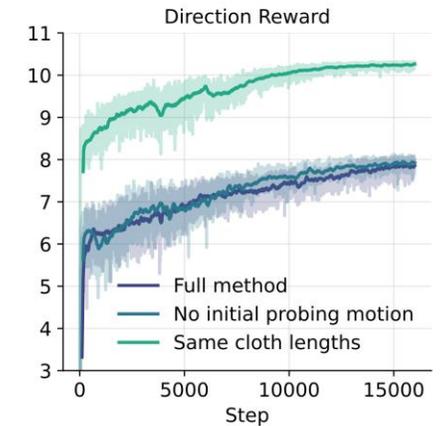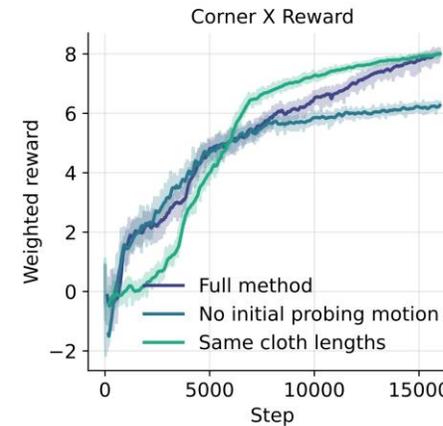
Conclusions on dynamically spreading cloth

- PPO + DMPs can be efficiently used to generate policies for fast dynamic bimanual robot motions in Isaac Sim
- Domain randomization enables the policy to generalize better to unseen cloths

- Randomization of cloth parameters (weight, size, material, …)
- Denser observations/rewards for multi-step manipulation (the agent can take into account the movement of cloth corners)
- Imitation learning
  - Generate a generic robot trajectory and train a policy to follow it
  - Perform reinforcement learning to refine the policy

ROMANDIC

Future research work @JSI

- More complex cloth manipulation actions (e. g., <u>imperfect grasp</u>, folding),

- Different tasks – folding into a box

- Combination with unfolding in the air...

- ... with imperfect grasp.

- Application for different domains (Flex-cycle)
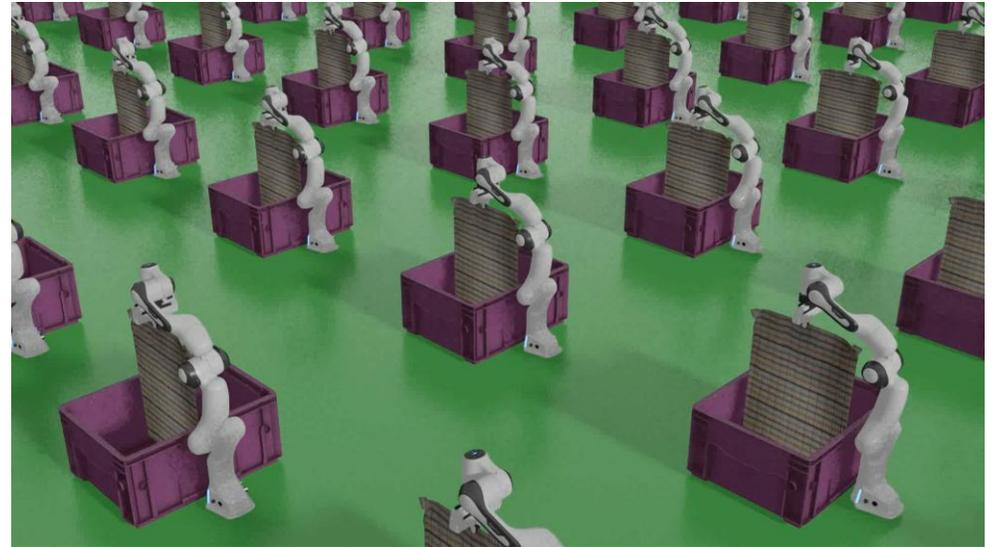
Dynamic folding into a box

Autonomous flattening

ROMANDIC

Autonomous flattening of garments

- Image-to-motion approach

- Deep neural network
  - Input: image
  - Output: trajectory of motion (or parameters of motion encoding)

- Tested on:
  - Encoder-decoder networks
  - Convolutional neural networks

35
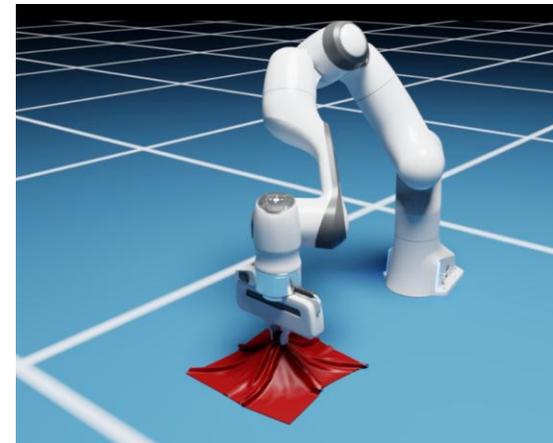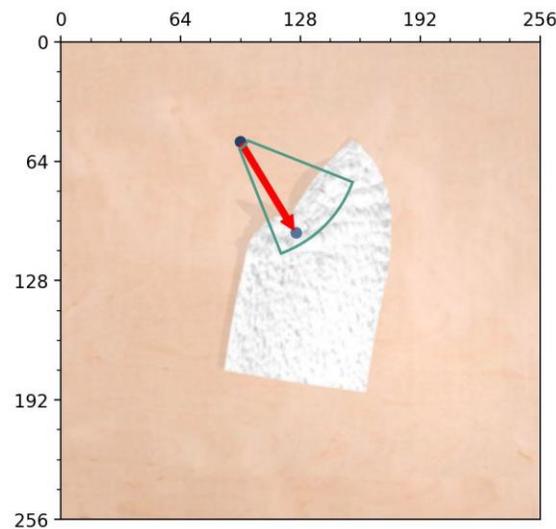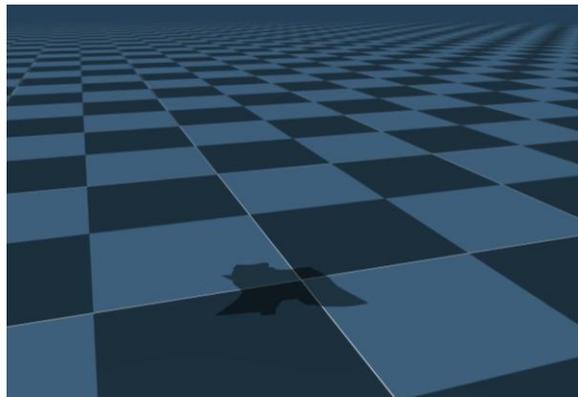
Network architecture

Autonomous flattening

d_{err} = 7.1 px; g_{err} = 2.39 px;

d_{err} = 3.89 px; g_{err} = 2.45 px;

d_{err} = 3.74 px; g_{err} = 1.45 px;
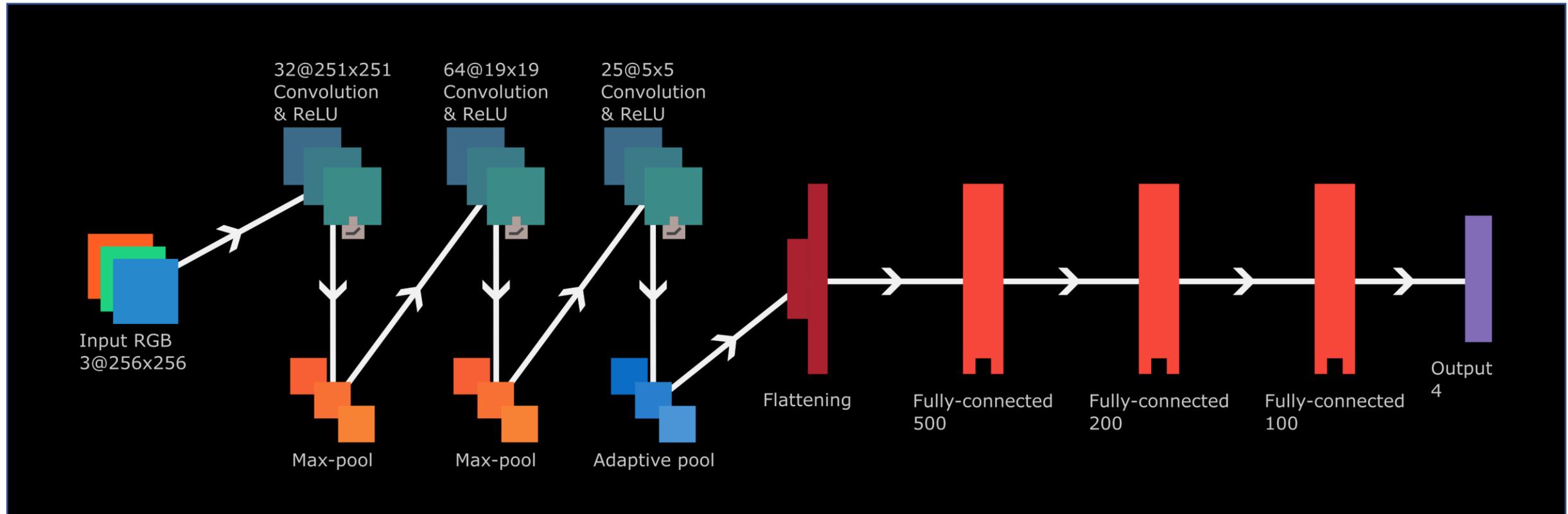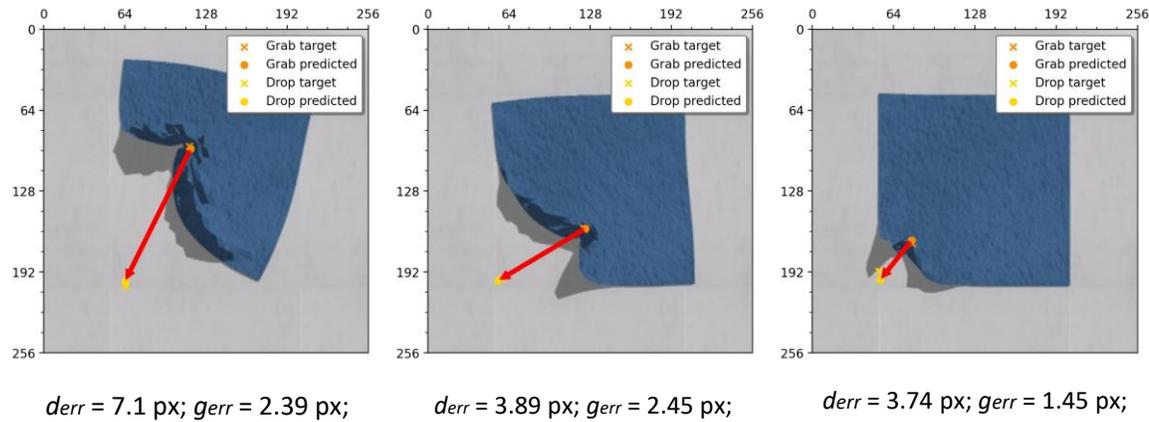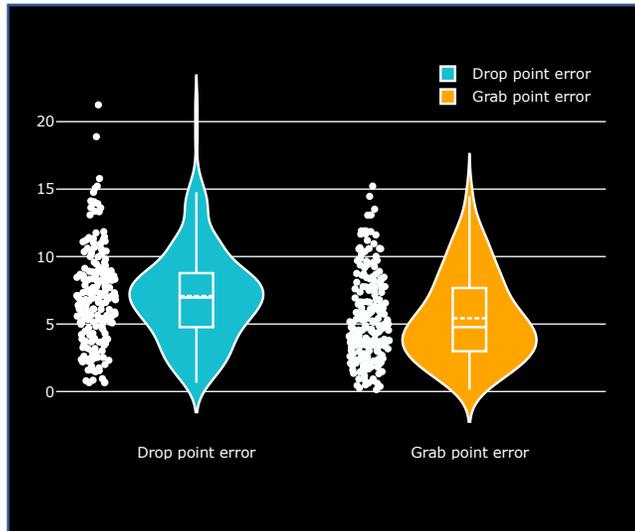
P. Nimac, M. Mavsar & A. Gams "Cloth Smoothing Simulation with Vision-to-Motion Skill Model." Zbornik 31. mednarodne Elektrotehniške in računalniške konference ERK 2022
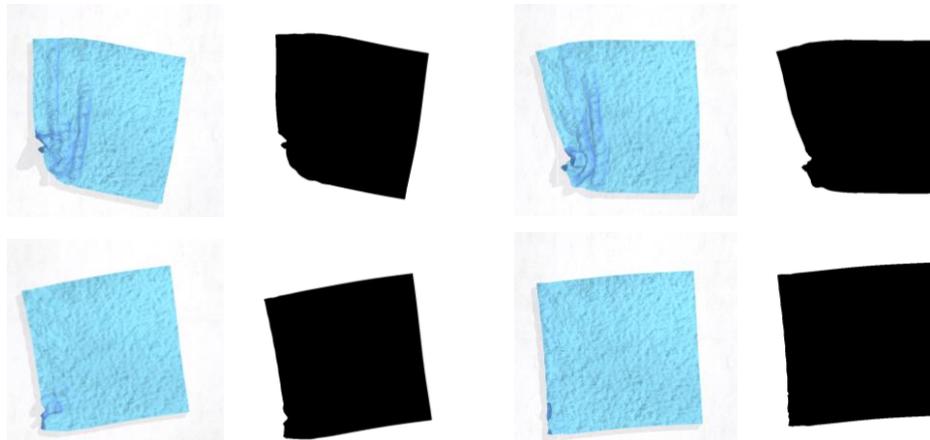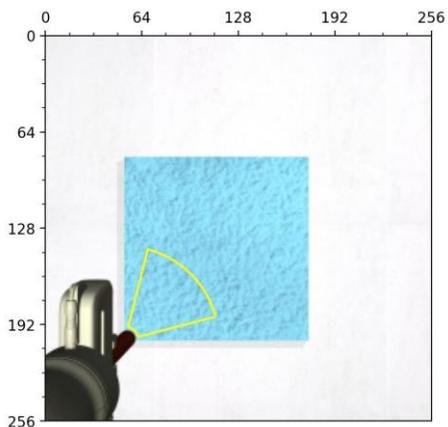
Nimac, Peter & Gams, Andrej "Cloth Flattening With Vision-to-Motion Skill Model." Advances in Service and Industrial Robotics. RAAD 2023

Autonomous flattening
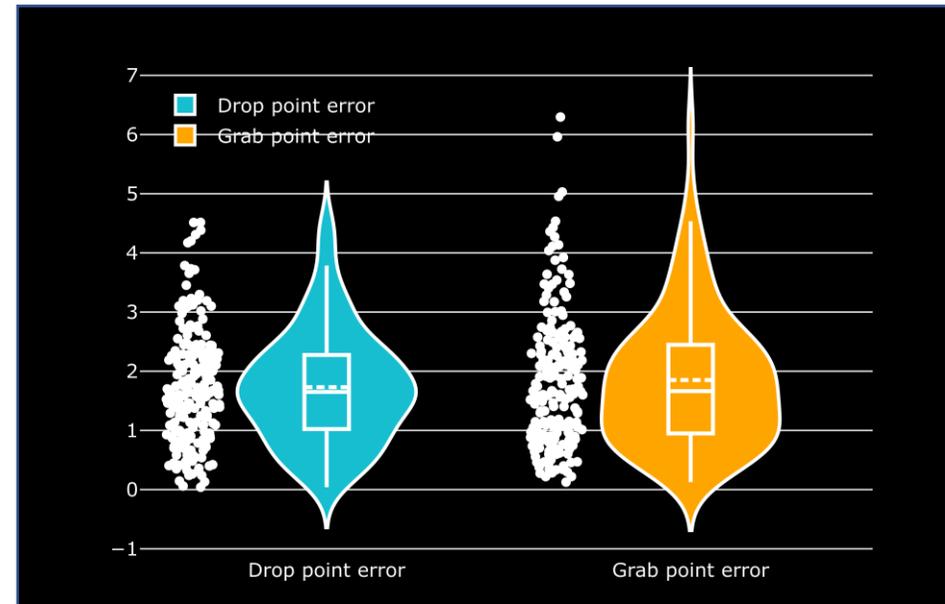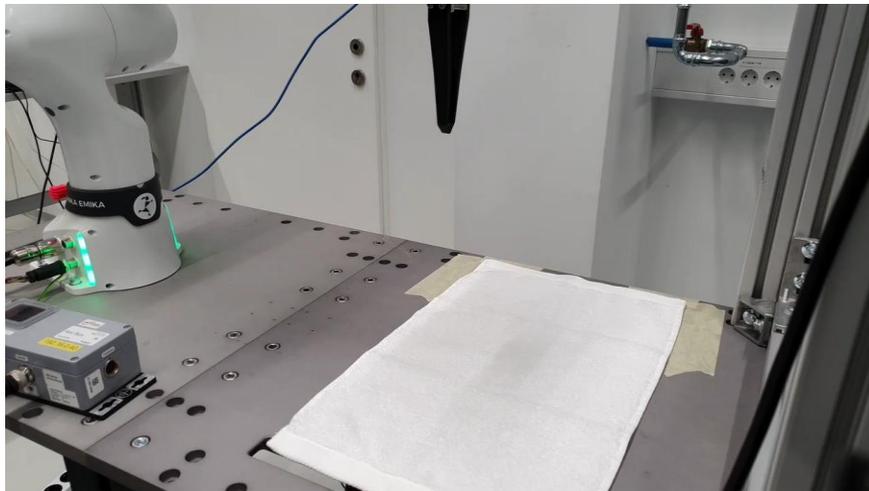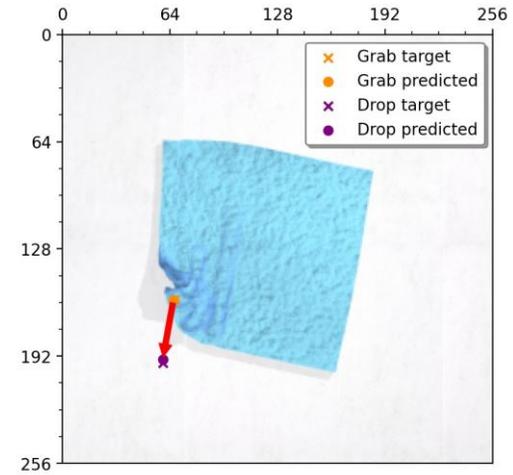
Bottom: Hard-coded demo

Sample quality

Model generalization
with high variability

Model generalization
with poor variability

Model generalization
with no variability

Desire to save:
- computation resources,
- computation time,
- data gathering (simulation) time.

Desire for our DNN to generalize as best as possible.

# Sample quality – grab and drop error
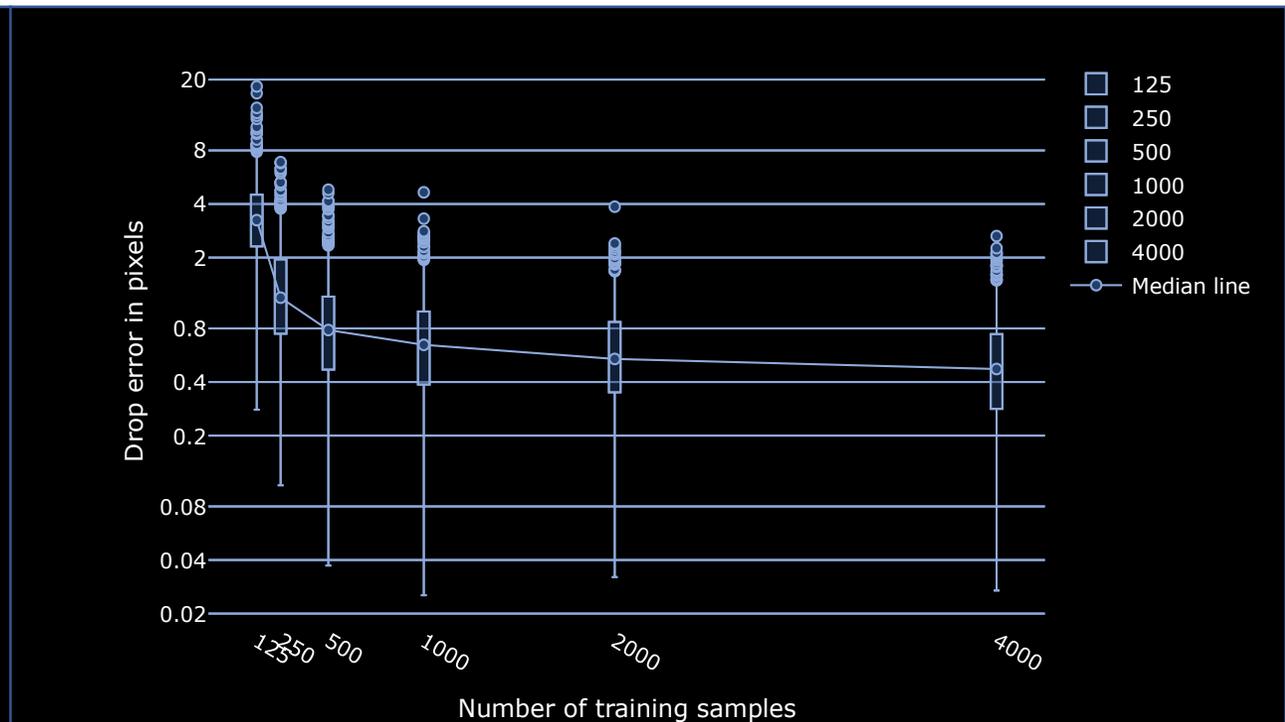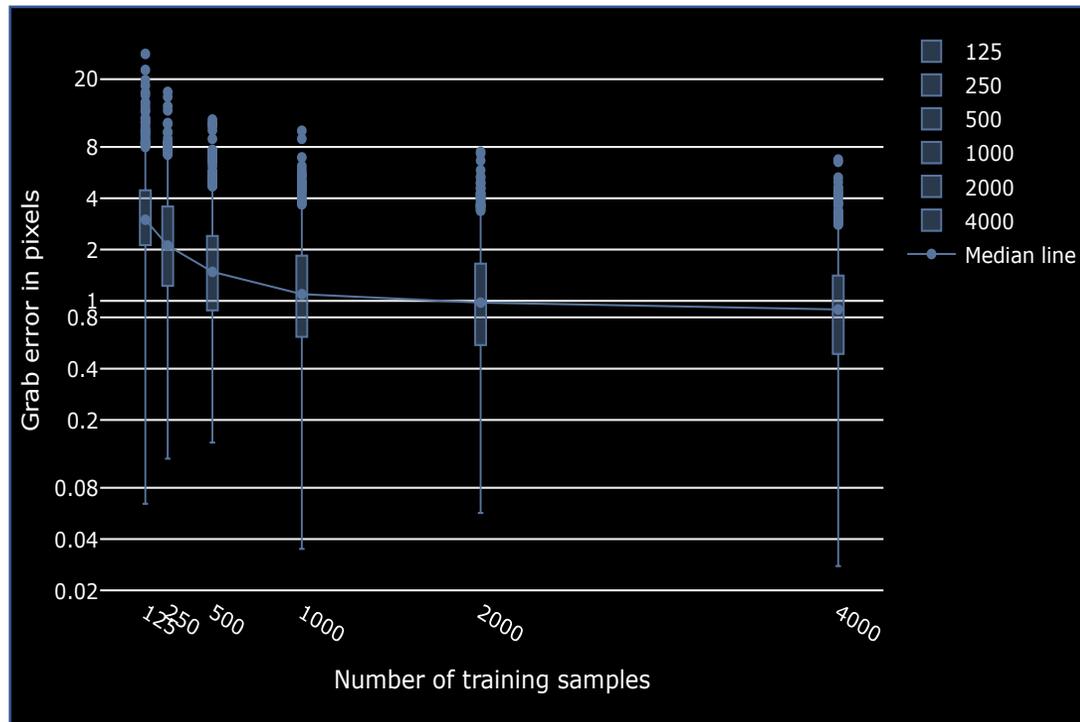
Real-world dataset and augmentations

- Flip (horizontal and vertical) – 50 %

- Shift, scale and rotation – 50 %
    - Shift in range (-0,0625; 0,0625)
    - Scale in range (-0,1; 0,1)
    - Rotation in range (-90; 90)

- Random resized crop – 100 %
    - Crop between 100 % and 15 % of original image
    - Resize to 256 x 256

- Color jitter – 50 %
    - Simulate jitter in hue, saturation and value

- Planckian jitter – 50 %
    - Simulate illumination changes

- Normalize
    - ImageNet mean values: (0,485; 0,456; 0,406)
    - ImageNet standard deviation : (0,229; 0,224; 0,225)
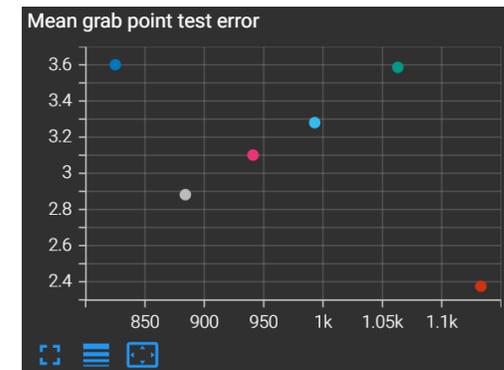
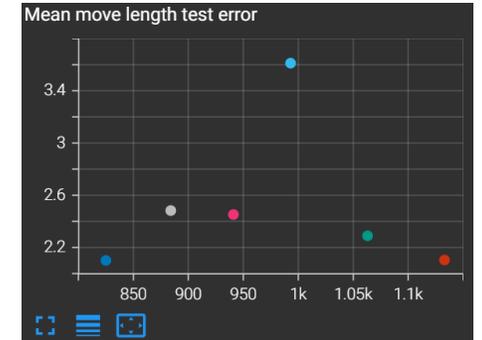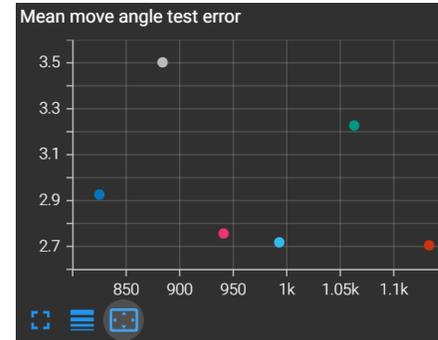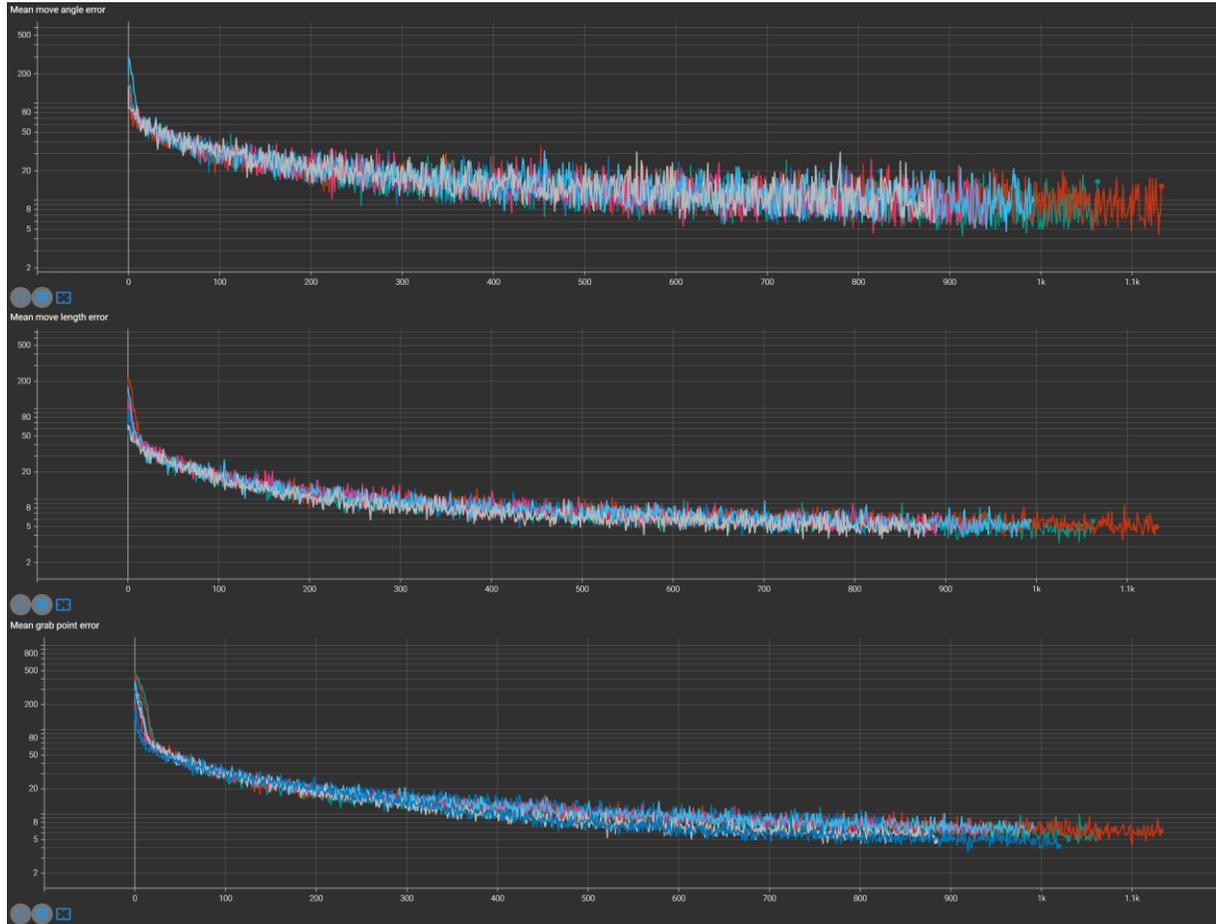- *Segmentation, pattern?*

With Albumentations library (https://albumentations.ai/)
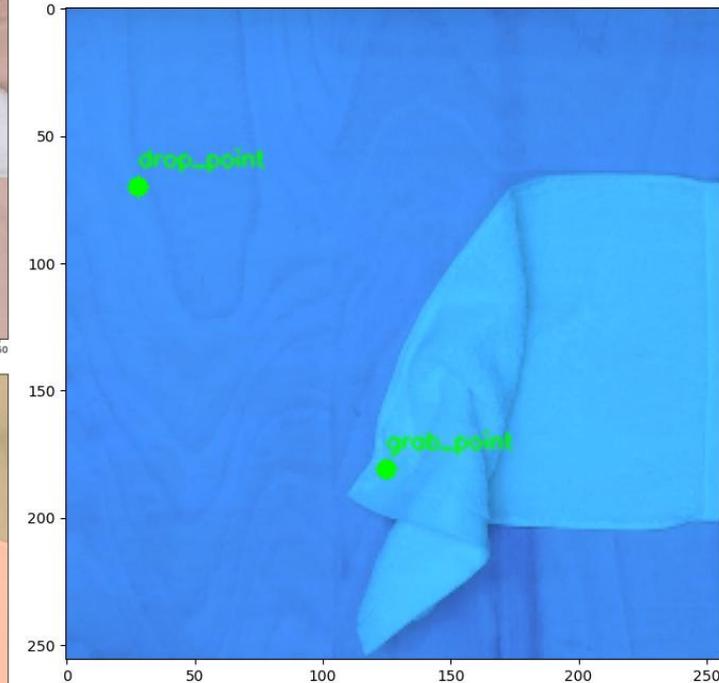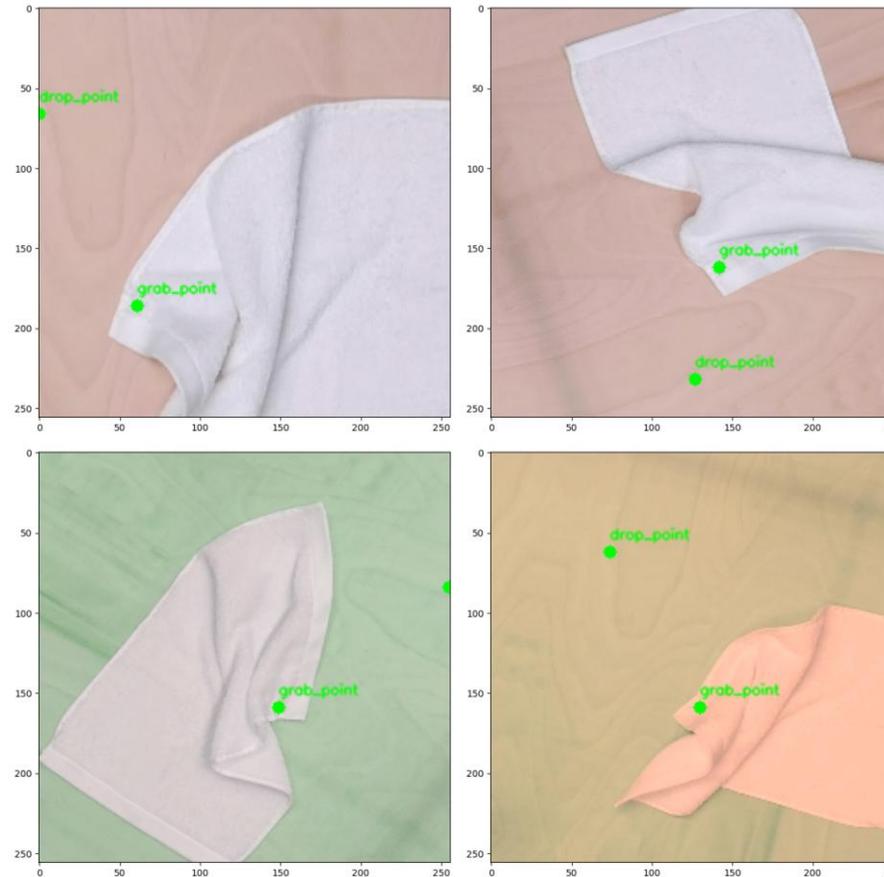
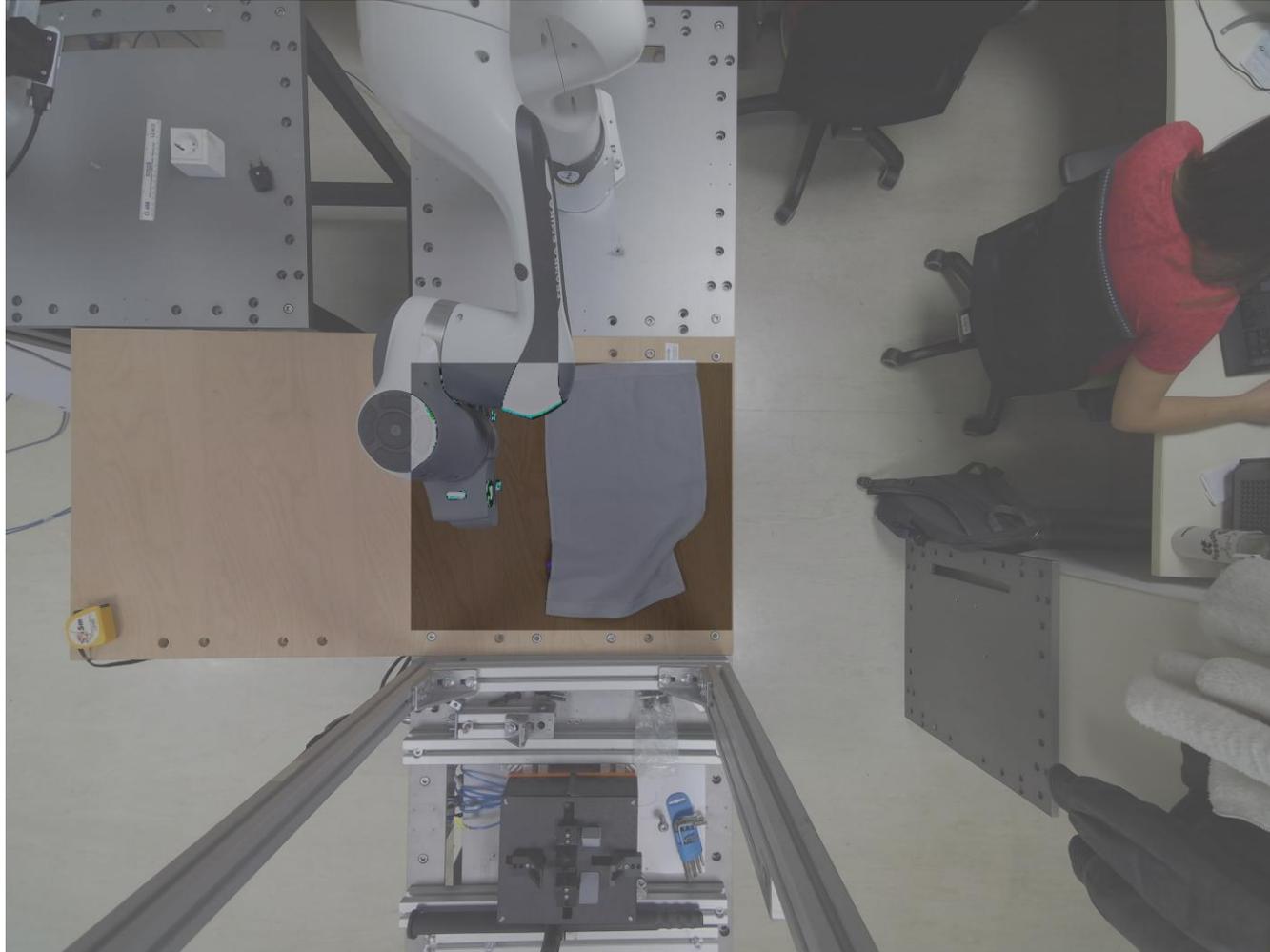# ResNET (Residual Network)
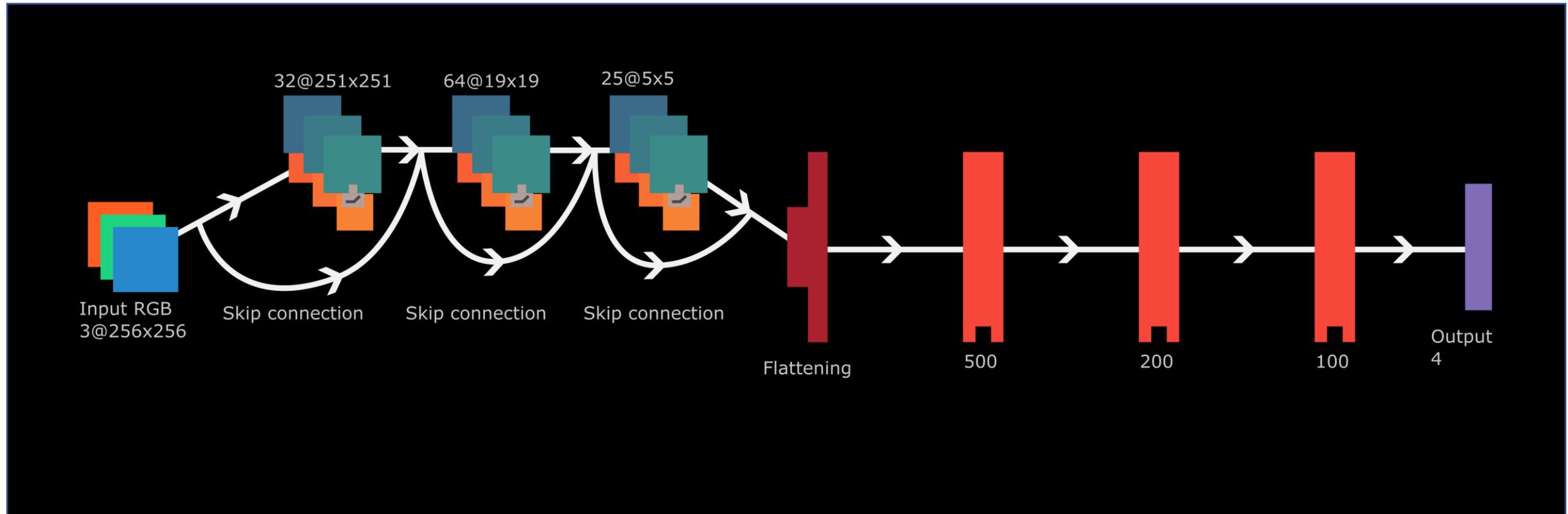
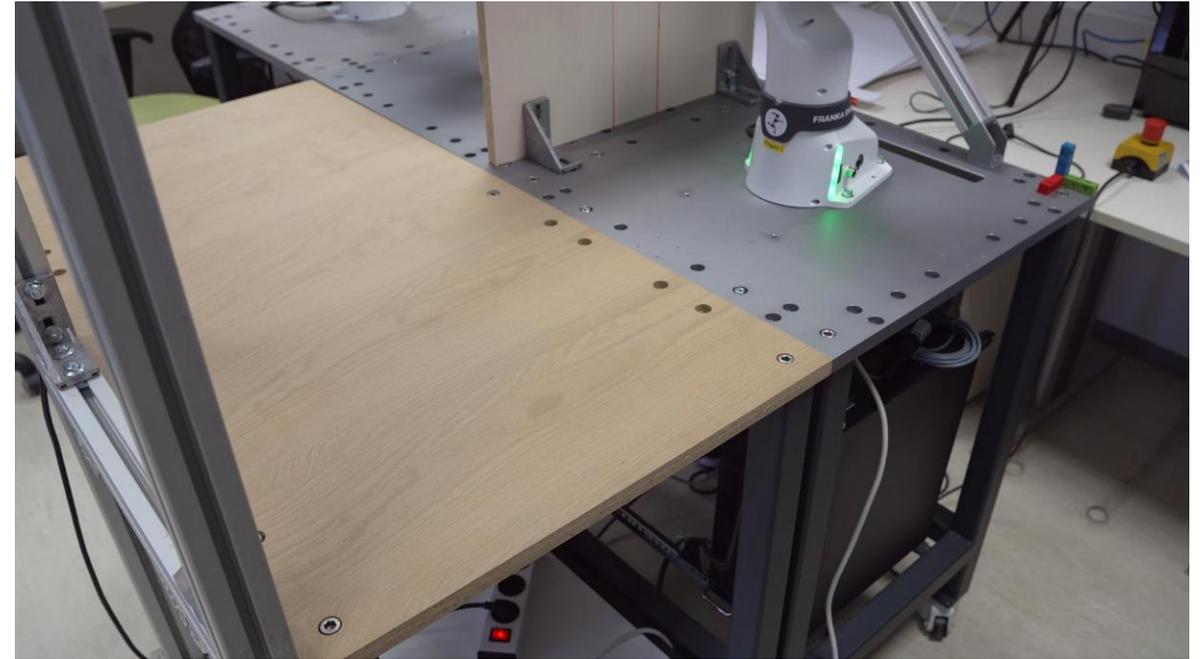
sketch

Training on real data

Real-world experimental setup

Setup

Real robot experiments

Conclusions and relation Hands-on exercises

- All three topics can be implemented with robots:
  - Unfolding in the air,
  - Dynamically spreading a cloth flat on the table,
  - Autonomous flattening.
- Presented solutions/approaches are one of the options, but results show they can be successfully implemented with robots.
- Other approaches exist.

In relation to the hands-on exercises:

- Deep reinforcement learning for flinging – dr. Matija Mavsar
- Data collection for flinging – Jan Jerićević & Mahed Dadgostar
- Image-to-motion – Peter Nimac

Collaborators in projects ROMANDIC and RTFM

Jožef Stefan Institute
- Peter Nimac
- Jan Jerićević
- dr. Matija Mavsar
- Prof. Aleš Ude

FRI-UL
- dr. Domen Tabernik
- Prof. Danijel Skočaj

ROMANDIC
Robot Manipulation of Deformables through Dynamic Actions

FRI | UNIVERZA V LJUBLJANI
Fakulteta za računalništvo in informatiko

JSI

CSIC
CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

KIT
Karlsruher Institut für Technologie

ROMANDIC

ARIS

The future is yours to shape!

PICK   SPREAD   FLATTEN
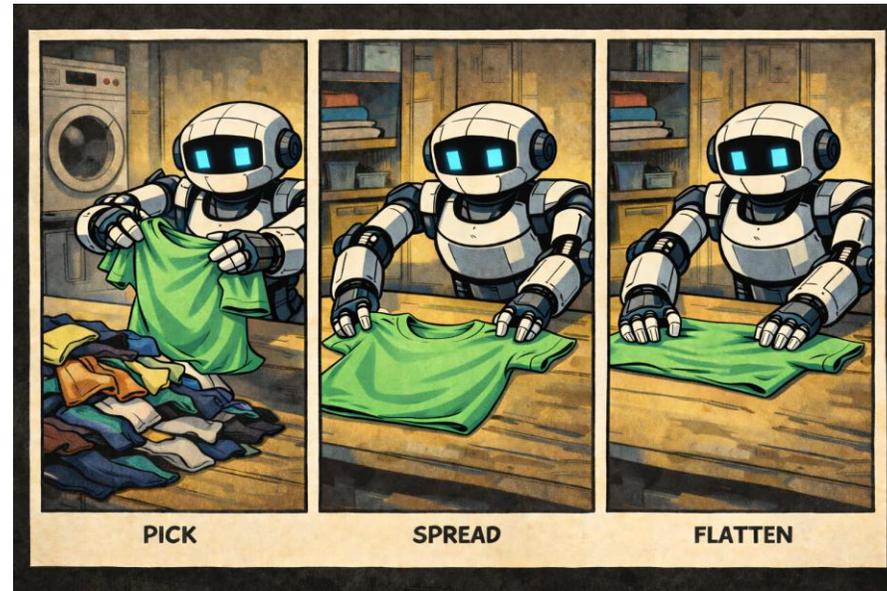
Thank you for your attention!