

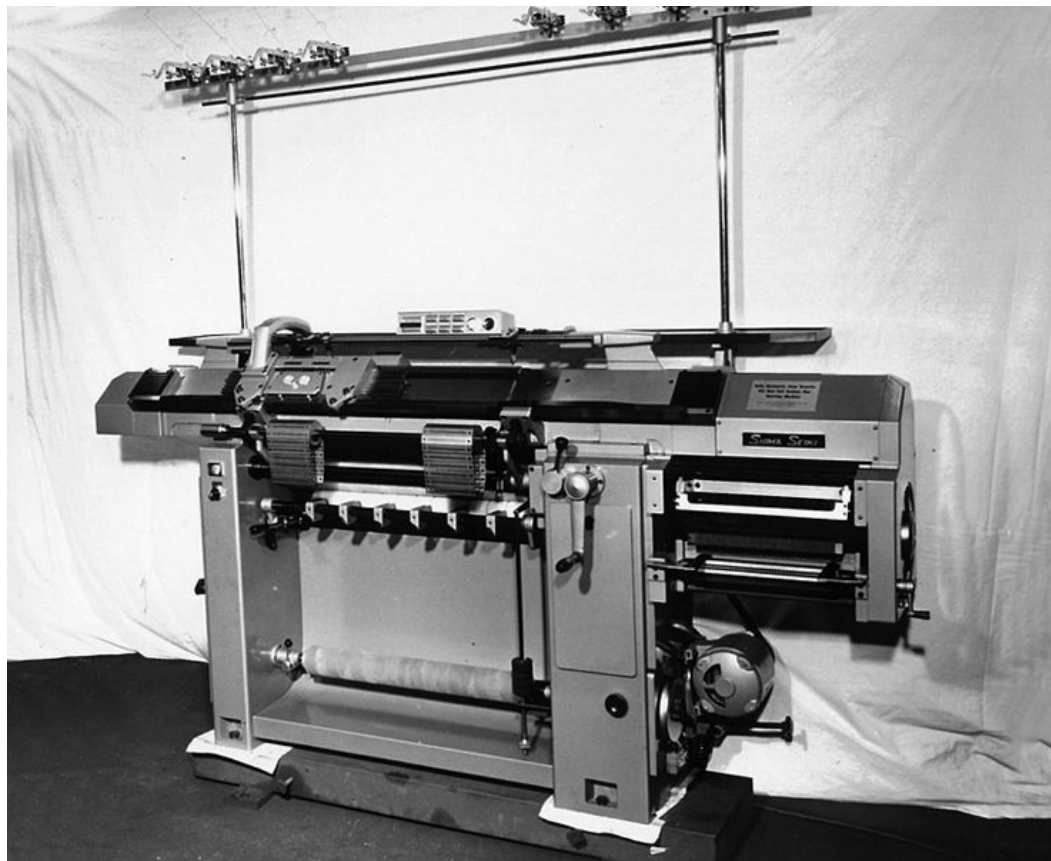
Customizing Textile and Tactile Skins for Interactive Industrial Robots

Bo

Knitted tactile skin



How it is made



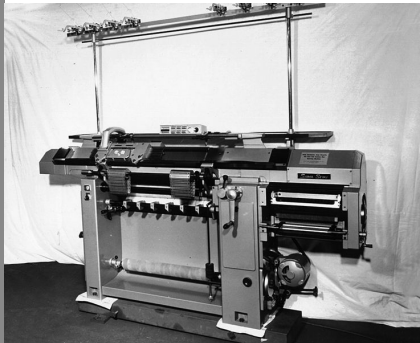
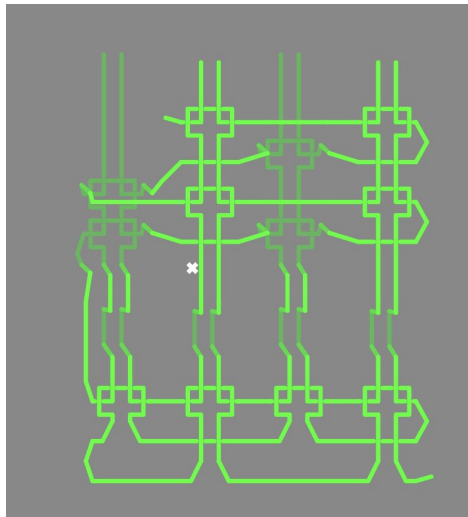
How it is made

```
for(var i = 0; i < NumHoles; i++) {
  currStripe = Math.floor(i/2);
  if (i==0) { // bottom_margin
    hole_locations_y.push(Math.floor(MarginHeight/2));
  }
  else if (i == NumHoles-1) { // top_margin
    hole_locations_y.push(MarginHeight+Math.floor(MarginH
  )
  }
  else if (i % 2) {
    hole_locations_y.push(MarginHeight+Math.floor(Conduct
  )
  }
  else {
    hole_locations_y.push(MarginHeight+Math.floor(Backgro
  )
  }
  hole_locations_x.push(start_row+space_row*i);
}

function knitMax2Min(pmin, pmax, carrier) {
  for (let n = pmax; n > pmin; --n) {
    if (n % 2 == front) {
      console.log("knit - f" + n + " " + carrier);
    }
    else {
      console.log("knit - b" + n + " " + carrier);
    }
  }
}

function knitMin2Max(pmin, pmax, carrier) {
  for (let n = pmin+1; n <= pmax; ++n) {
    if (n % 2 == front) {
      console.log("knit + f" + n + " " + carrier);
    }
    else {
      console.log("knit + b" + n + " " + carrier);
    }
  }
}
```

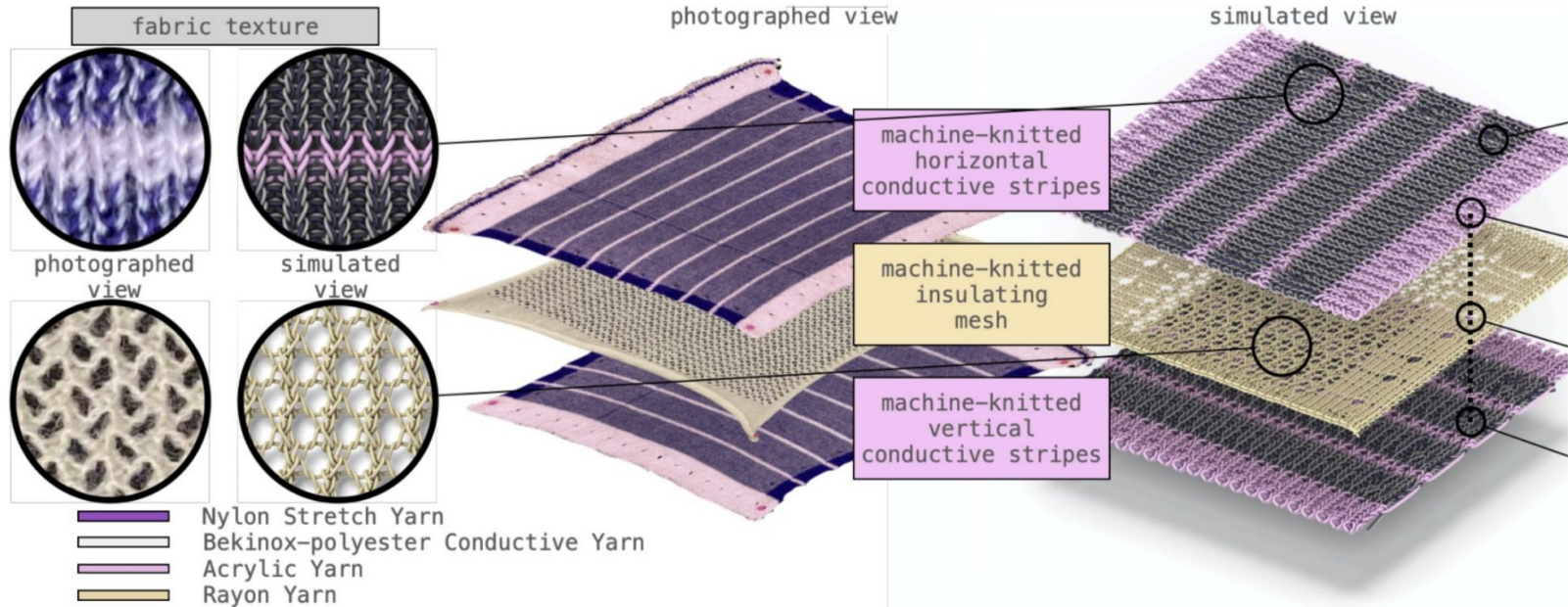
```
;!knitout-2
;;Machine: SWGN2
;;Carriers: 1 2 3 4 5 6 7 8 9 10
inhook 2
x-stitch-number 101
tuck - f20 2
tuck - f18 2
tuck - f16 2
tuck - f14 2
tuck - f12 2
tuck - f10 2
tuck - f8 2
tuck - f6 2
tuck - f4 2
tuck - f2 2
tuck + f1 2
tuck + f3 2
tuck + f5 2
tuck + f7 2
tuck + f9 2
tuck + f11 2
tuck + f13 2
tuck + f15 2
tuck + f17 2
tuck + f19 2
knit - f20 2
knit - f19 2
knit - f18 2
knit - f17 2
```

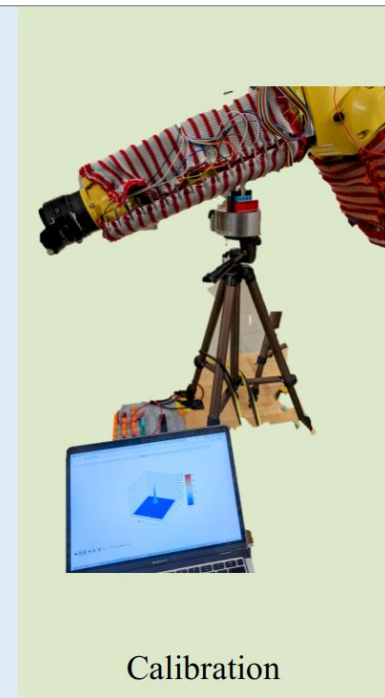
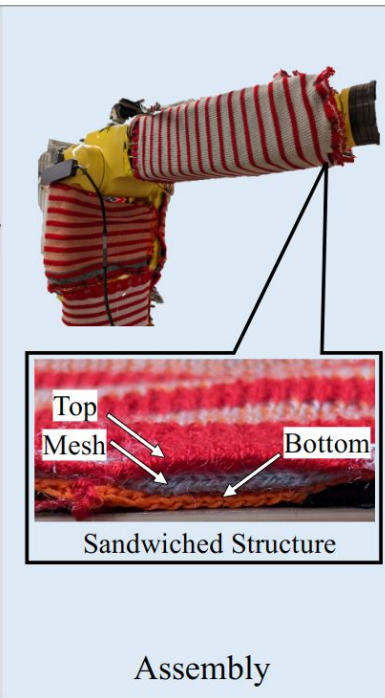
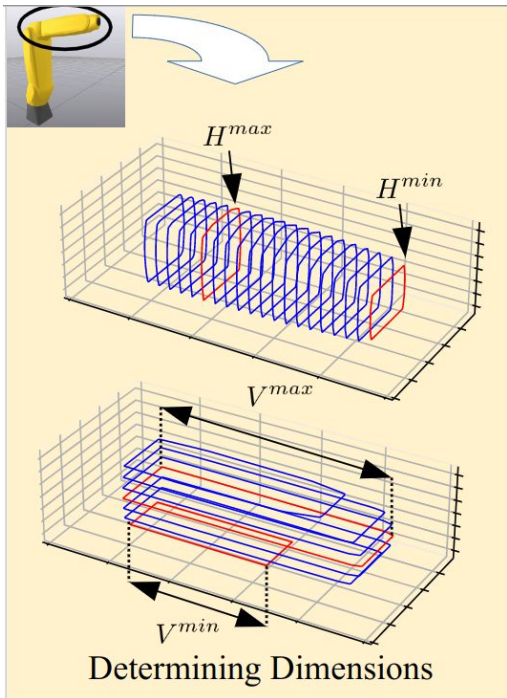


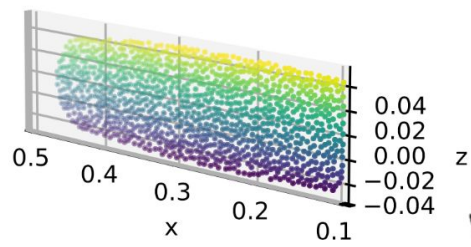
Bo ✂

How it is made

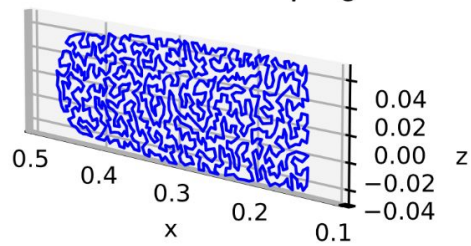
conductive stripes



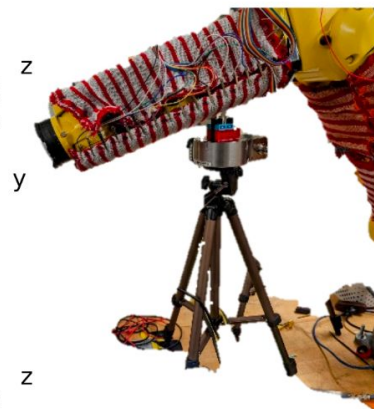




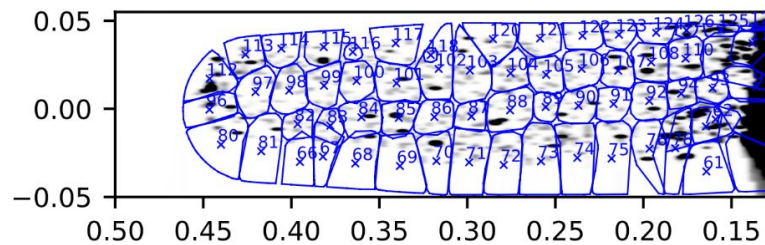
Surface Sampling



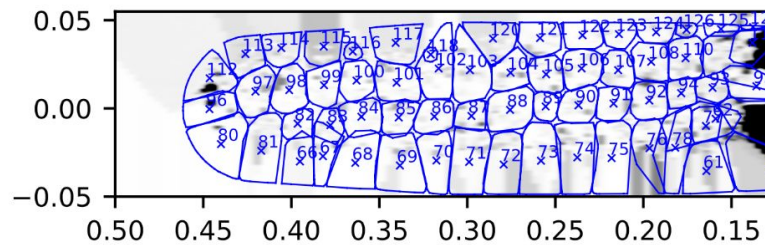
Trajectory Planning



Calibration Run



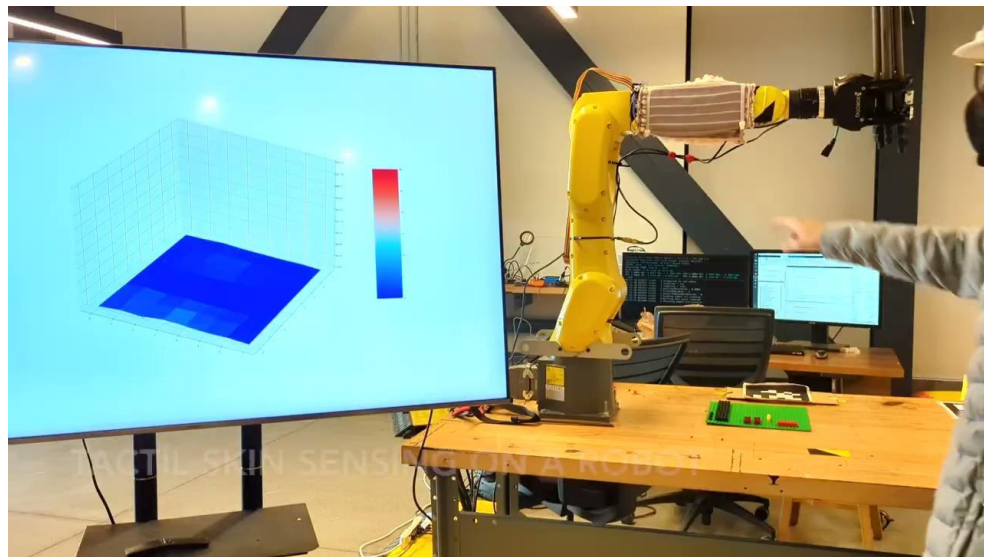
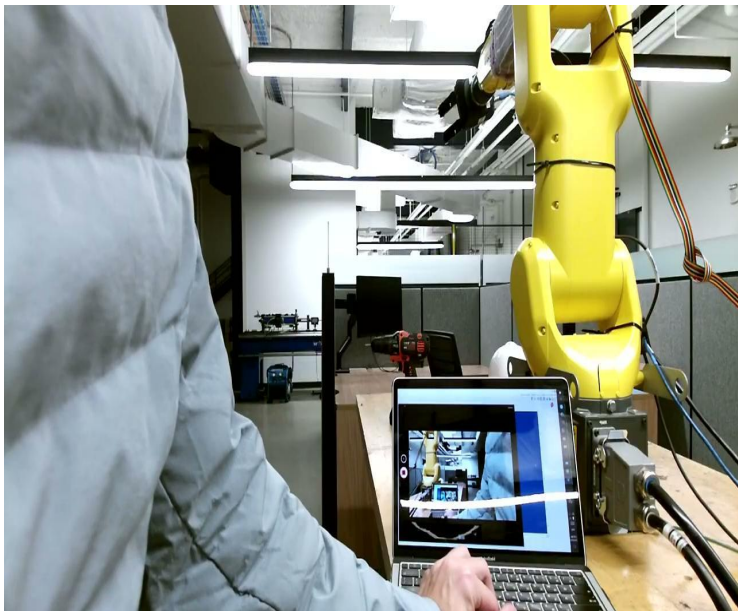
Before Force Calibration



After Force Calibration

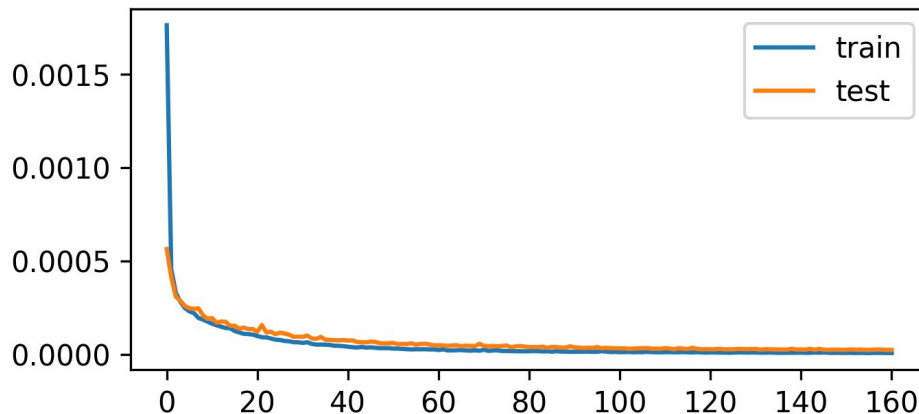
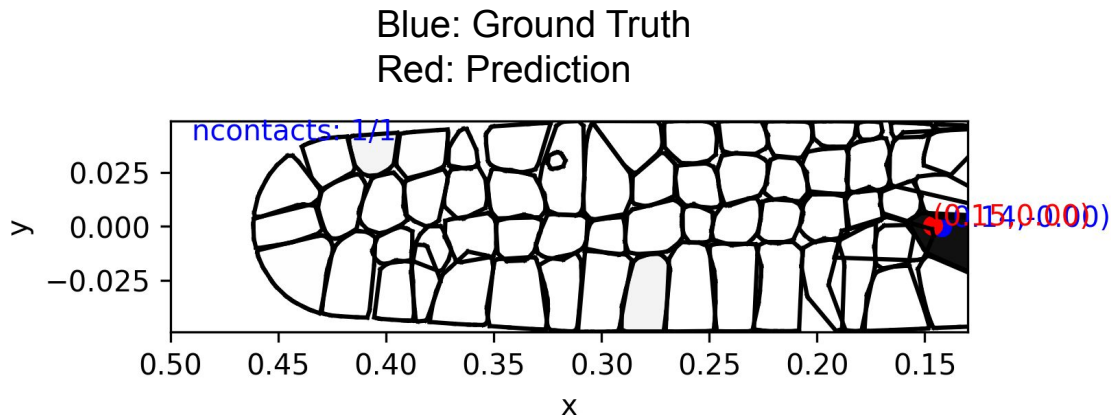
Signal Preprocessing

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_0 \right\}$$

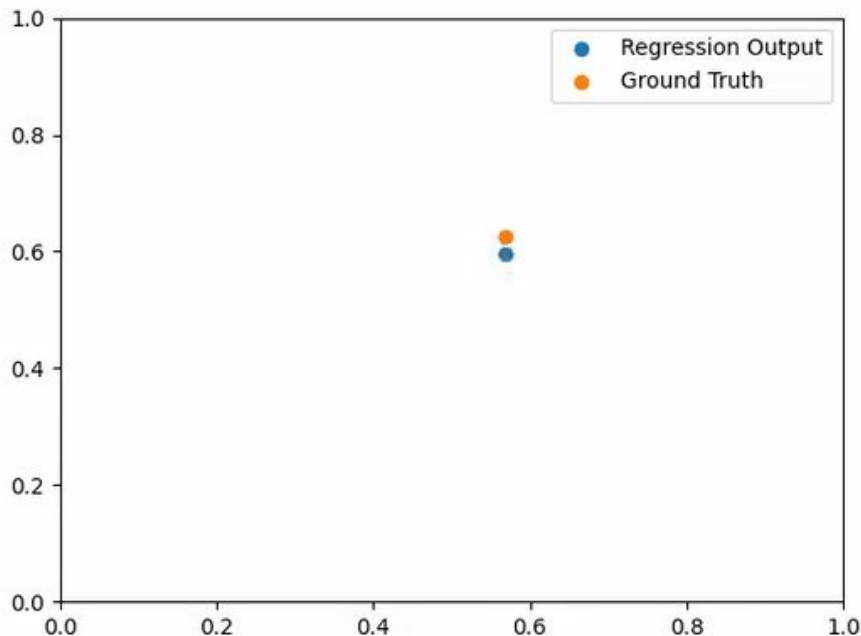
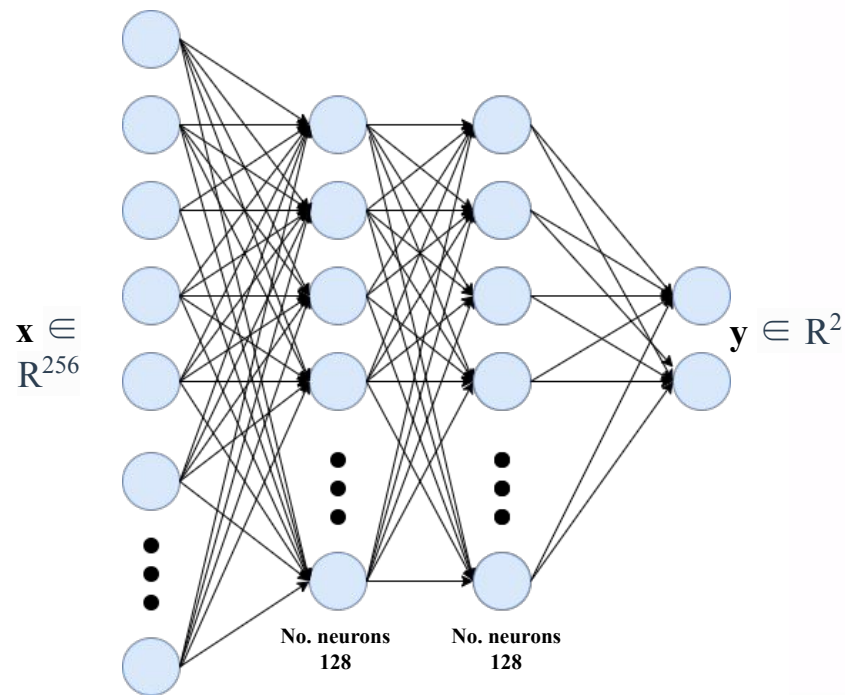


Single-touch Model

- Architecture
 - 4 linear layers, 2 hidden
 - Input dimension: 256 nodes
 - Output dimension: 2 nodes
 - Epochs: 300
 - Learning Rate: .001
 - Loss Function: MSE
 - Optimizer: Adam
- Final results:
 - Train Loss: $1.9e-5$
 - Test Loss: $3.18e-5$

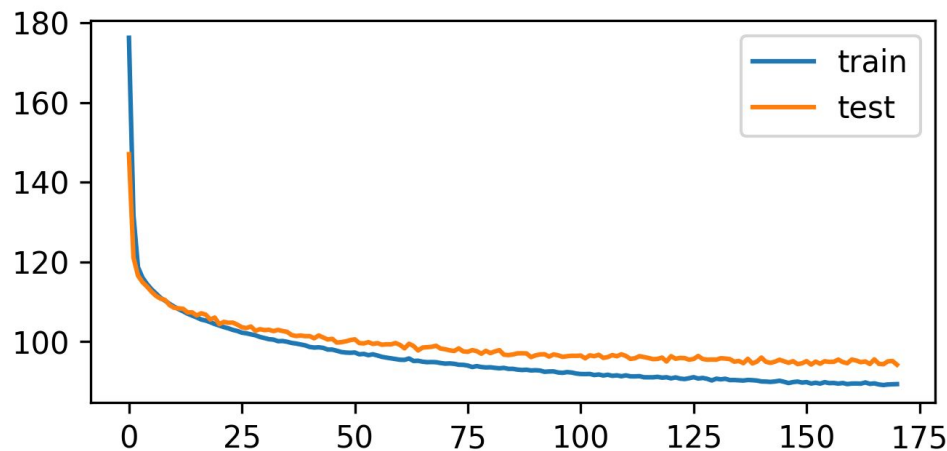
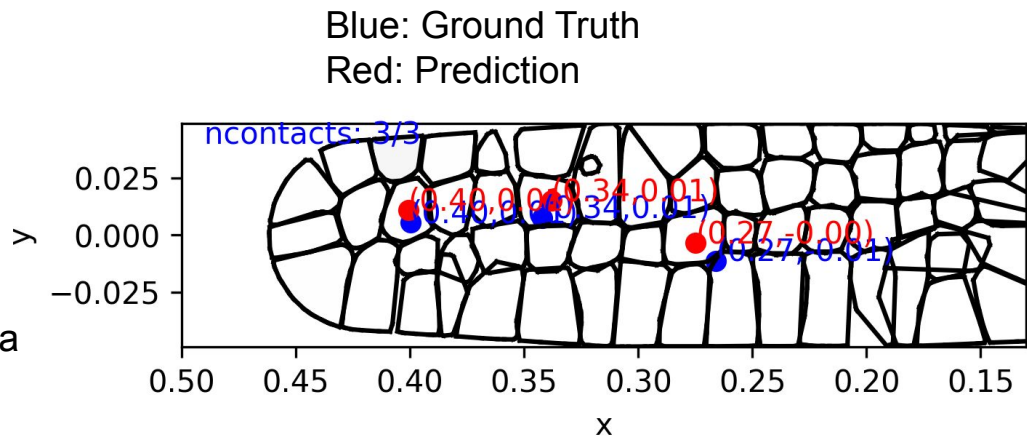


Localization Results using Single touch MLP

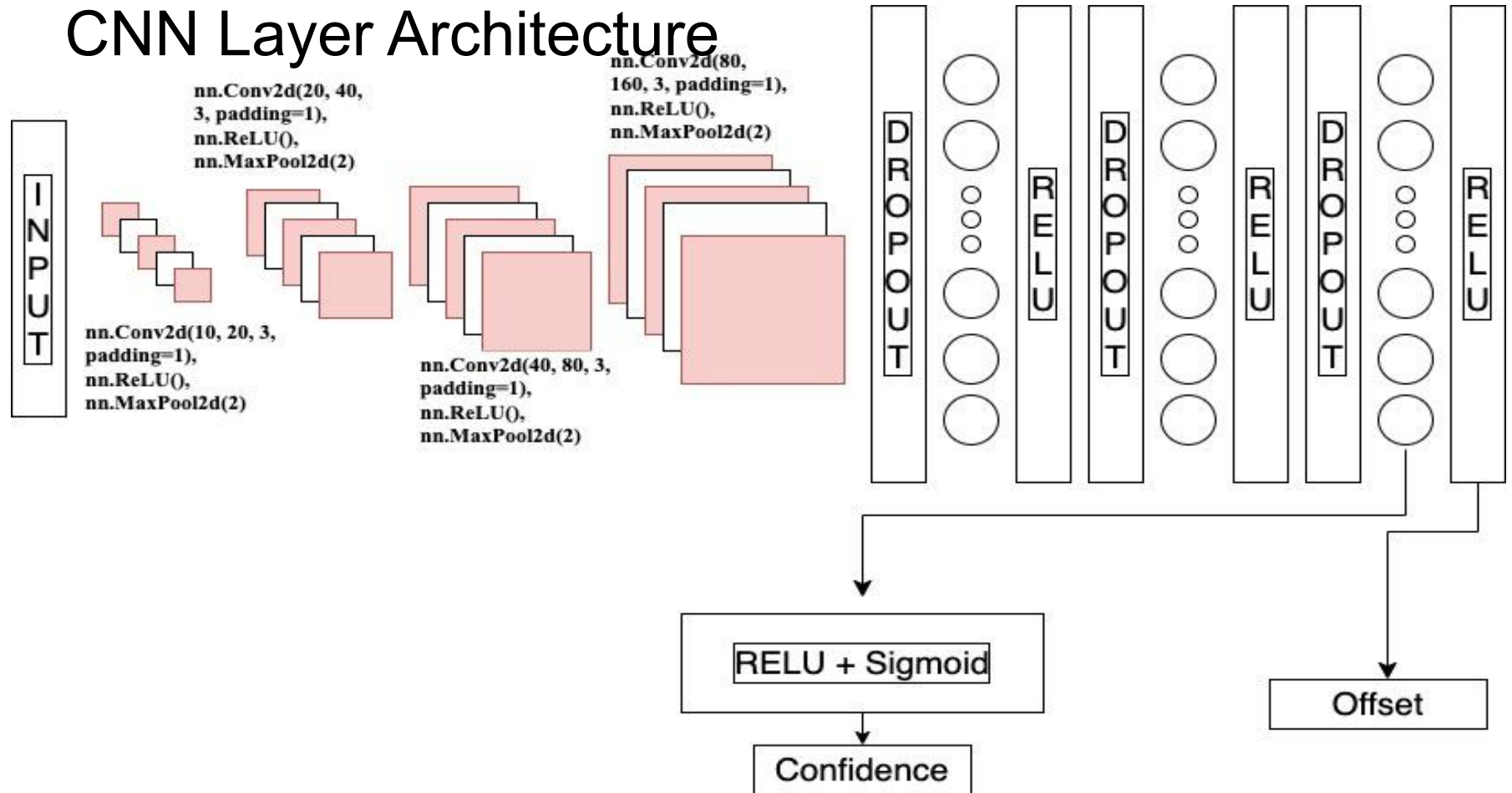


Multi-touch Model

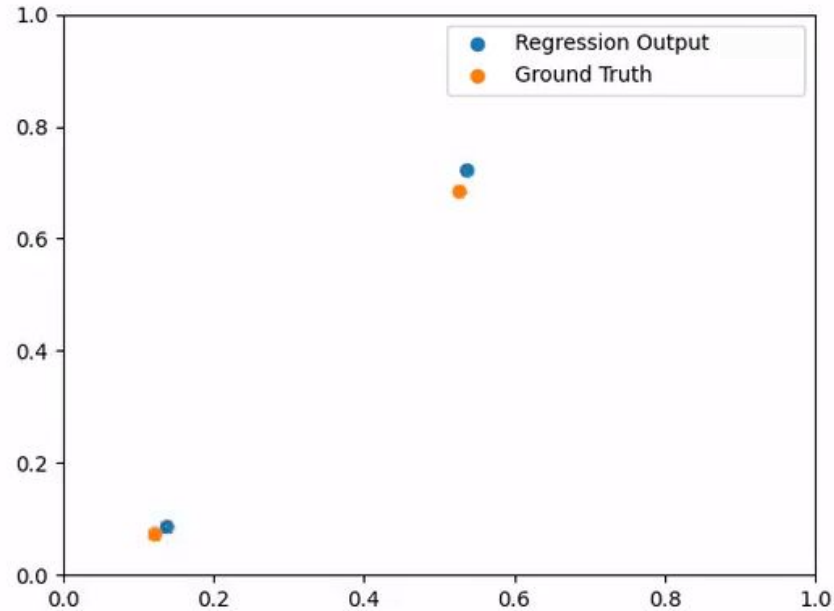
- Data
 - Superposition on single-point data
- Architecture
 - CNN, Maxpool, Linear
 - Epochs: 175
 - Learning Rate: .001
 - Loss Function: MSE
 - Optimizer: Adam
- Anchors
 - Centroid of the k-Nearest-Neighbor classifier prediction on random samples
- Final results:
 - Train Loss: 89.1874
 - Test Loss: 94.9838



CNN Layer Architecture



Localization Results using Multi-touch MLP



Demonstration (Multi-Touch)

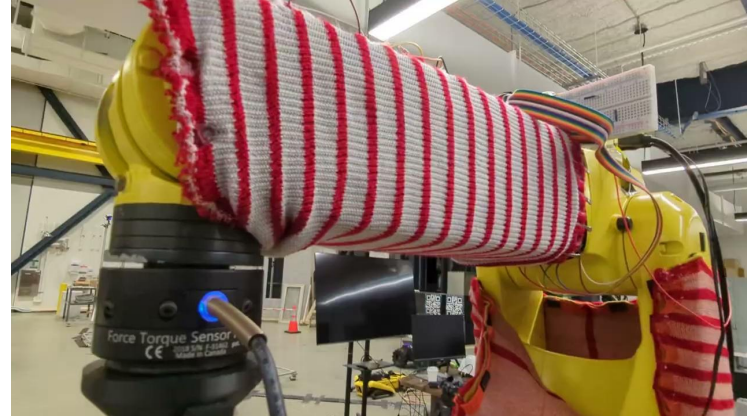
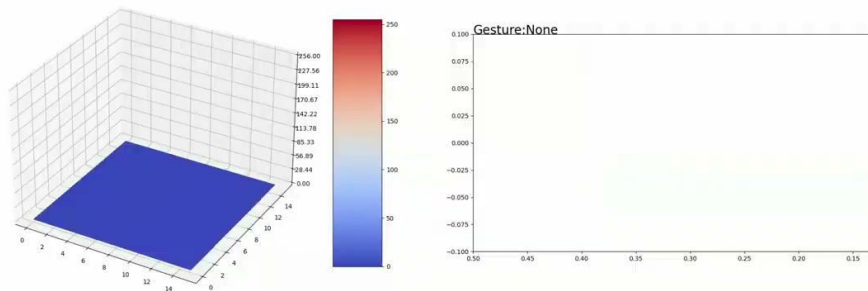


Figure 1



Admittance Control



Obstacle Avoidance

$$\begin{aligned} \min_{\dot{q}_{\text{cmd}}} \quad & \|\dot{x}_{\text{lead}} - J(q)\dot{q}_{\text{cmd}}\|_2^2 \\ \text{s.t.} \quad & u_{\text{rep}}^T J_{\text{rep}} \dot{q}_{\text{cmd},n} \geq 0 \end{aligned}$$

Original motion



Avoid Motion

