MAKING SENSE OF VISION AND TOUCH: SELF-SUPERVISED LEARNING OF MULTIMODAL REPRESENTATIONS FOR CONTACT-RICH TASKS

PAPER BY: MICHELLE A. LEE, YUKE ZHU, KRISHNAN SRINIVASAN, PARTH SHAH, SILVIO SAVARESE, LI FEI-FEI, ANIMESH GARG, JEANNETTE BOHG

PRESENTED BY: MYKHAILO KOSHIL

25.11.2021

CONTENT

1. Introduction
2. Applications
3. Approaches
4. Paper presentation

1. INTRODUCTION [1]

- *Assembly task* is the process of putting manufactured pieces together in some predefined order.
- *Assembly motion* is a motion of the manipulator that moves a part into an assembled, i.e. into a required spatial arrangement or into contact with the other part.
- Assembly usually involves high precision and low tolerance between parts ➡ uncertainties in sensing and control are not trivial to handle

[1] Hägele M., Nilsson K., Paires J.N. (2008) Industrial Robotics. In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_43

1. INTRODUCTION

- *Compliant motion* is a motion, that is constrained by the contact between the held part and another part in the environment.
- Reduces uncertainty ➡ simplifies task ➡ used a lot in assembly
- Usually, more than one such motion is required in an assembly task
- *Peg-in-hole* insertion task is among the most used such motions, and the topic of today’s paper

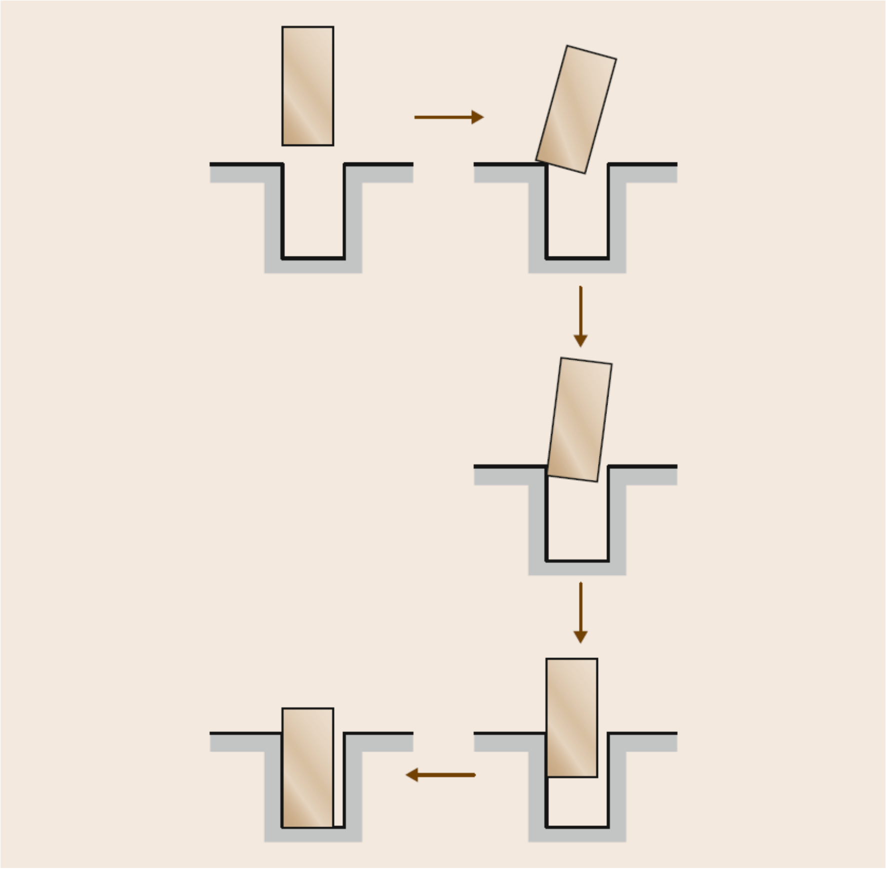


Fig.1: Contact transitions while executing peg insertions

Fig.1 from Hägele M., Nilsson K., Paires J.N. (2008) Industrial Robotics. In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_43

1. TASK FORMULATION

- *Contact state* – can be defined topologically as a set of primitive contacts, each of which is defined by a pair of contacting surface elements in terms of faces, edges, and vertices.
- *Peg insertion* – a compliant motion that consist of combing two mating parts into predefined contact state.

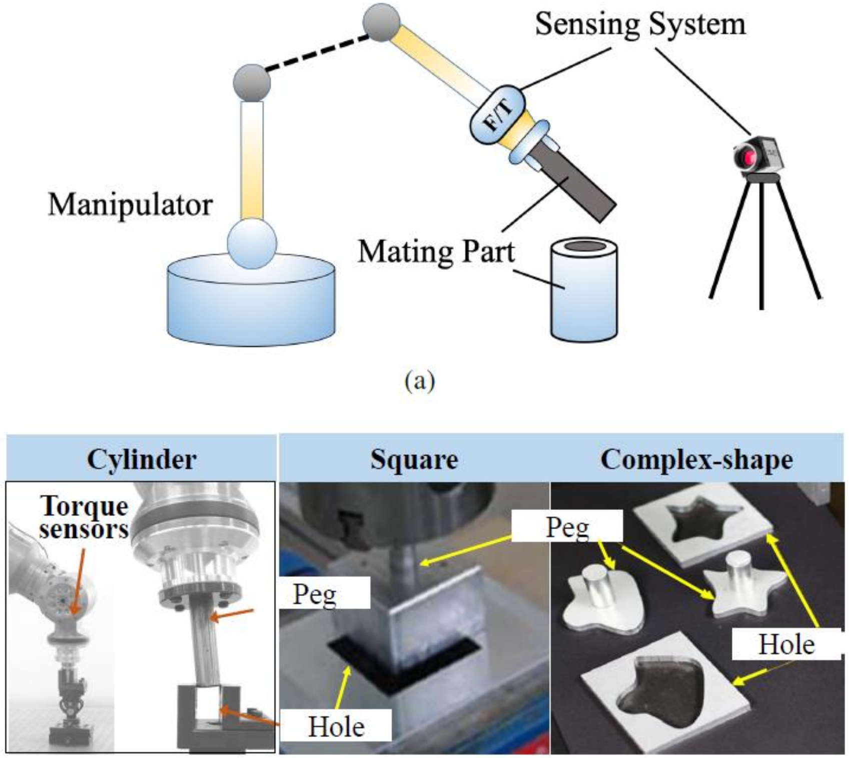
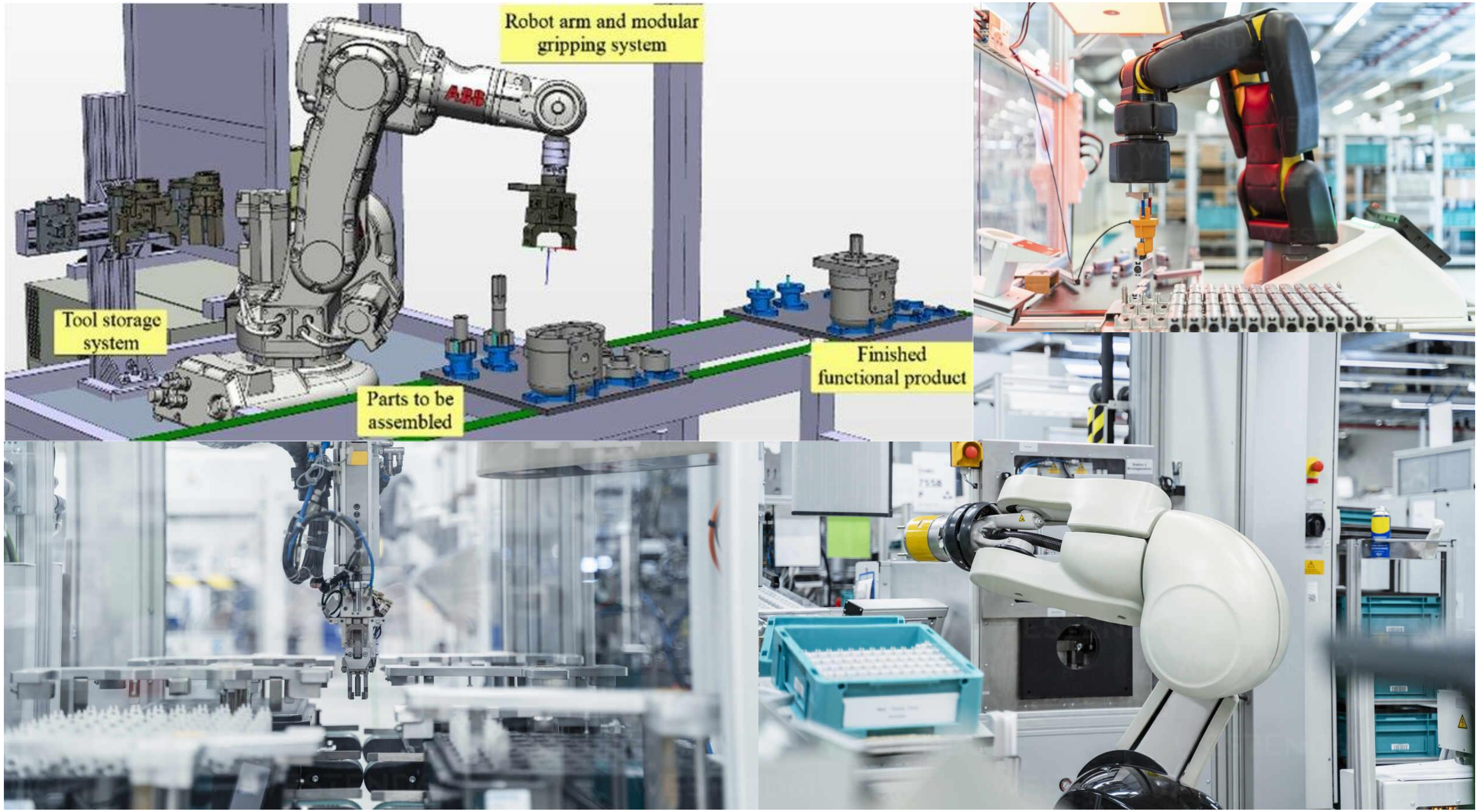


Fig.2[2]: Setup of the robotic peg-in-hole assembly system

[2] Xu, Jing & Hou, Zhimin & Liu, Zhi & Qiao, Hong. (2019). Compare Contact Model-based Control and Contact Model-free Learning: A Survey of Robotic Peg-in-hole Assembly Strategies.

2. APPLICATIONS


Robot arm and modular gripping system
Tool storage system
Parts to be assembled
Finished functional product

https://www.westend61.de/en/imageView/DIGF07198/arm-of-assembly-robot-functioning-inside-modern-factory-stuttgart-germany
https://www.researchgate.net/figure/A-typical-setup-of-a-robotic-assembly-cell_fig1_350789609

3. APPROACHES

- Two main strategies to deal with compliant motions[1]:
  - Passive Compliance: incorporation of the compliant motion for the error correction during the assembly motion, without the need of an active and explicit recognition and reasoning of contact states between parts.
    - Example: RCC (mechanical device)
  - Active Compliance: error correction is based on online identification or recognition of contact states in addition to feedback of contact forces.
    - Allows for broader range of assembly tasks with large uncertainties and tasks beyond assembly where compliance is required.

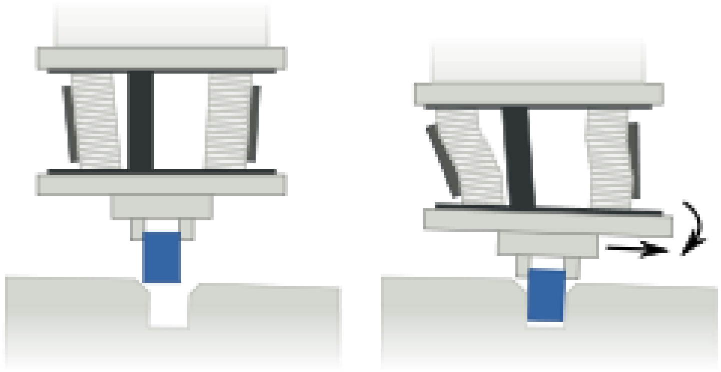


Fig.3: Remote compliance center (RCC)
source:
https://upload.wikimedia.org/wikipedia/commons/thumb/d/d1/Remote_Center_of_Compliance.svg/1200px-Remote_Center_of_Compliance.svg.png

[1] Hägele M., Nilsson K., Paires J.N. (2008) Industrial Robotics. In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg. p. 914 https://doi.org/10.1007/978-3-540-30301-5_43

3. APPROACHES

- Classical strategies (model based) need to be preprogrammed by experts using domain knowledge.[1]
  - Expensive and time consuming
  - Sensitive to the configuration of the working space
    Robot working space for the assembly task needs to be structured
  - Not adaptive
  - This creates difficulties and increases cost
- Learning-based methods are popular now, because of their potential to overcome these issues. It is also possible to incorporate prior knowledge or expert demonstrations in order to speedup learning[2].

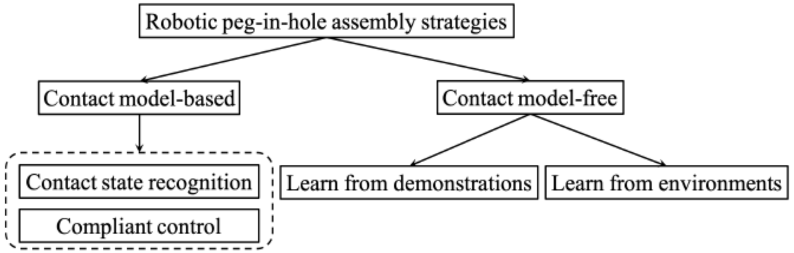

Robotic peg-in-hole assembly strategies
Contact model-based
Contact model-free
Contact state recognition
Compliant control
Learn from demonstrations
Learn from environments

Fig.4: classification of assembly strategies [2]

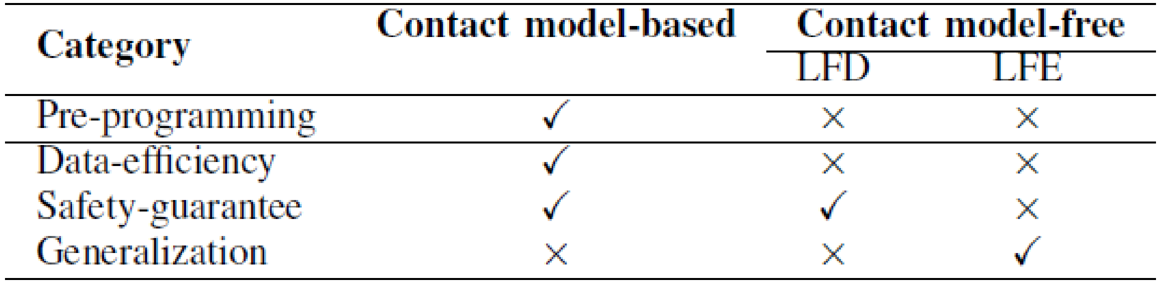
[2] Xu, Jing & Hou, Zhimin & Liu, Zhi & Qiao, Hong. (2019). Compare Contact Model-based Control and Contact Model-free Learning: A Survey of Robotic Peg-in-hole Assembly Strategies.

3. APPROACHES

Learning based methods can be divided further:

1. Learning form demonstration (LFD)
   Seems to be not so popular [2] in context of the peg-in-hole tasks.
2. Learning from the environment (LFE)
   One of the most widely [2] used approaches when it comes to manipulation tasks and offering advantage of generalization to unseen task.

Main problem with learning-based approaches is low data efficiency.



Category	Contact model-based	Contact model-free	
		LFD	LFE
Pre-programming	✓	×	×
Data-efficiency	✓	×	×
Safety-guarantee	✓	✓	×
Generalization	×	×	✓

Fig.5: comparison of the different assembly strategies [2]

[2] Xu, Jing & Hou, Zhimin & Liu, Zhi & Qiao, Hong. (2019). Compare Contact Model-based Control and Contact Model-free Learning: A Survey of Robotic Peg-in-hole Assembly Strategies.

4. PAPER PRESENTATION

4.1 PAPER SUMMARY

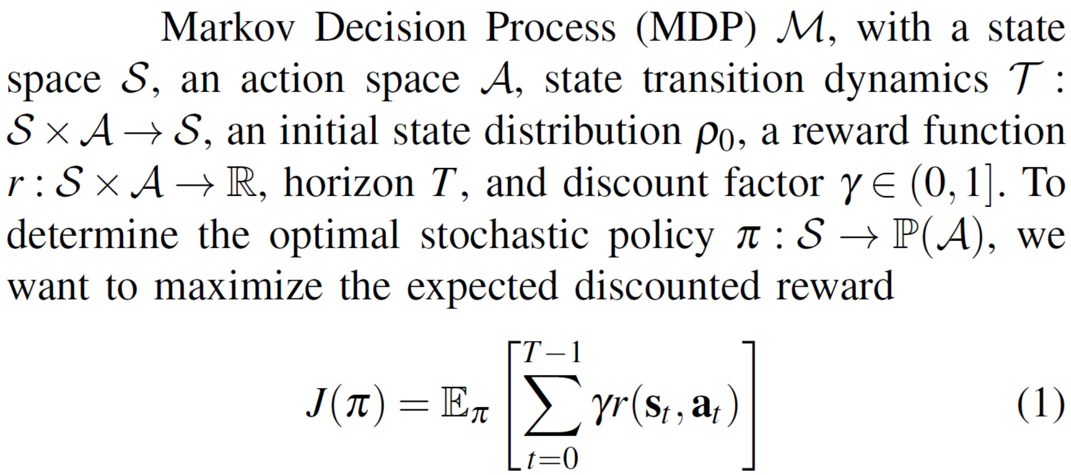
- Title: Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks[3]
- Authors: Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, Jeannette Bohg
- Employer: Department of Computer Science, Stanford University,
  A. Garg is also at Nvidia, USA
- Submitted on: 24 Oct 2018
- ICRA 2019 best paper award video and interview

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.1 PAPER SUMMARY

- Focus of the paper
  - Peg-in-whole task, representation of the multimodal data
- What is special about it
  - Apparently, it works in real life
  - Use of 3 modalities: vision, sensory input and proprioception
  - Use of self-supervision to acquire the training data
- Goals
  - Create concise representation for the multimodal data that can describe the current state of the system, and can be used for solving manipulation tasks i.e., used as an input for controller
  - To learn a policy on a robot for a manipulation task
  - Evaluate the impact of modalities and ability to transfer representations for different tasks

4.2 PROBLEM STATEMENT



Markov Decision Process (MDP) $\mathcal{M}$, with a state space $\mathcal{S}$, an action space $\mathcal{A}$, state transition dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, an initial state distribution ρ_0, a reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, horizon T, and discount factor $\gamma \in (0,1]$. To determine the optimal stochastic policy $\pi : \mathcal{S} \rightarrow \mathbb{P}(\mathcal{A})$, we want to maximize the expected discounted reward

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma r(\mathbf{s}_t, \mathbf{a}_t) \right] \qquad (1)$$

- Manipulation task modeled as a finite-horizon, discounted Markov Decision Process (MDP),
- Goal – learn the policy $\pi\colon \mathcal{S} \rightarrow \mathbb{P}(\mathcal{A})$
- Policy represented by the neural network θ_π
- S will be the learned representation from the high dimensional input
- A is defined over continuously-valued, 3D displacements Δx in Cartesian space

4.4 MODEL FOR REPRESENTATION LEARNING

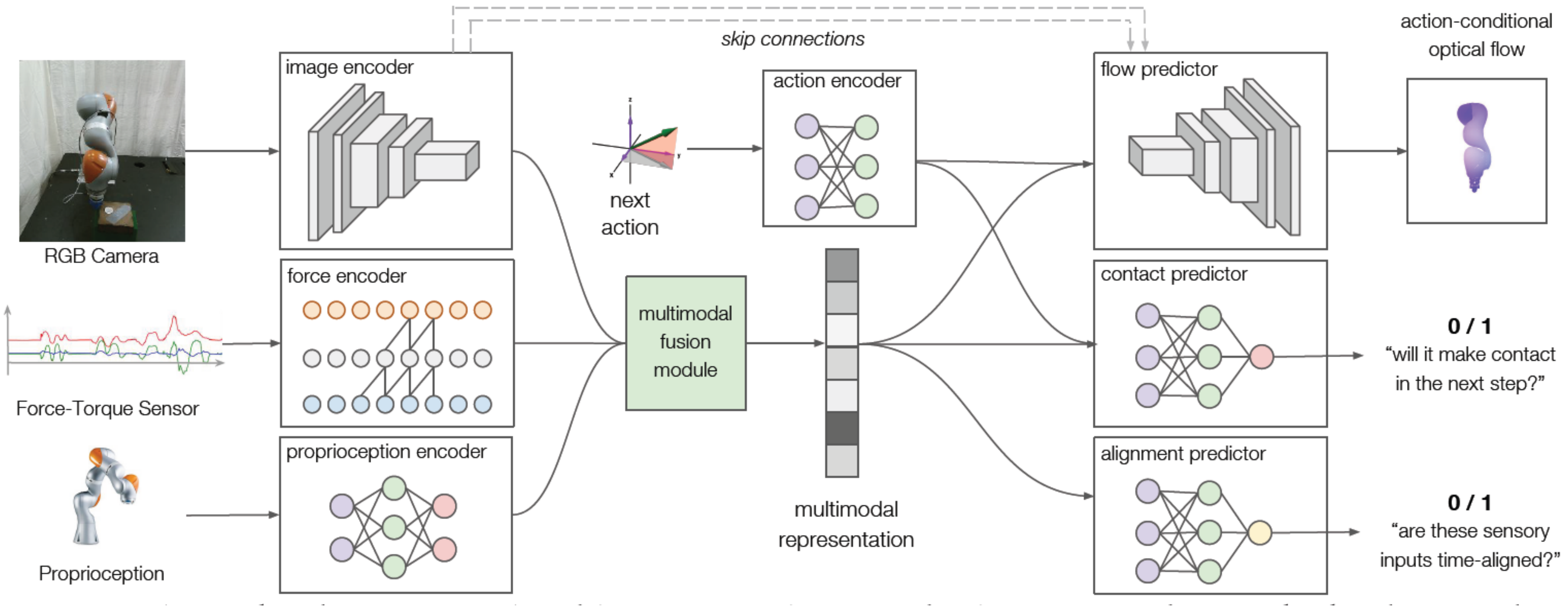


Fig. 6: model for the representation learning [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.4 MODEL FOR REPRESENTATION LEARNING: MODALITY ENCODERS

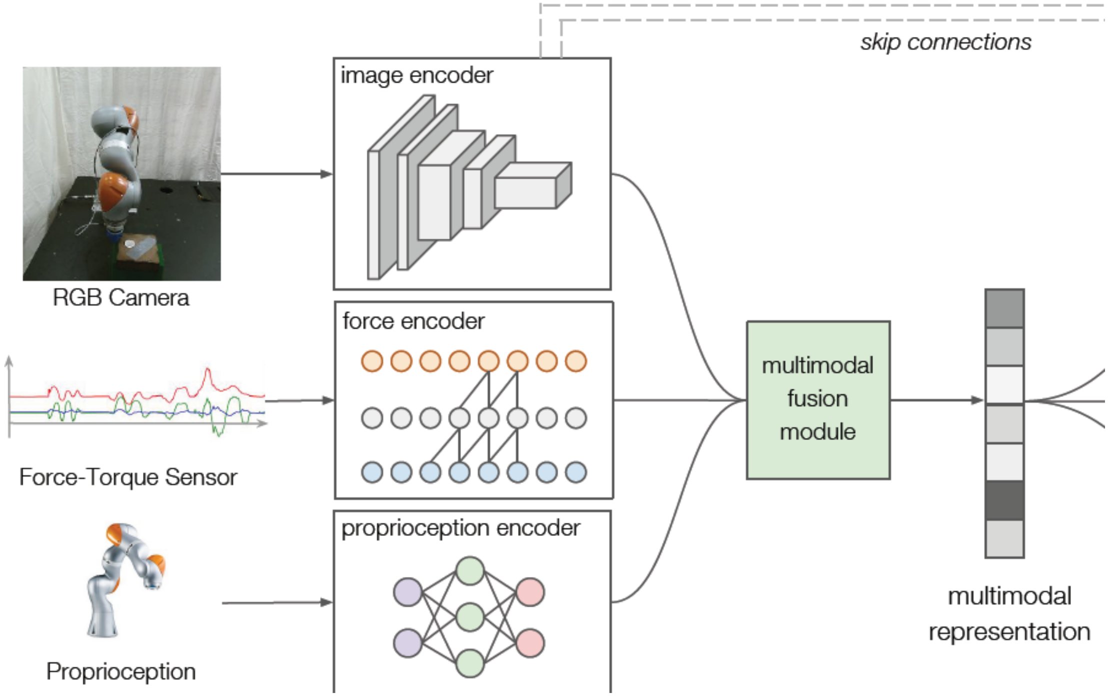


Fig. 6: model for the representation learning [3]

- Images: 128x128 RGB images => 6-layer CNN, similar to the FlowNet + fully connected layer => 128-d feature vector
- Haptic feedback: last 32 readings of 6-axis F/T sensor as a 32x6 time series => 5-layer causal convolutions with stride 2 => 64-d feature vector
- Proprioception: current position and velocity of TCP => 2-layer MLP => 32-d feature vector
- Multimodal representation: concatenation of the 3 vectors above => 2-layer MLP => 128-d multimodal representation

4.4 MODEL FOR REPRESENTATION LEARNING: SELF-SUPERVISED PREDICTIONS

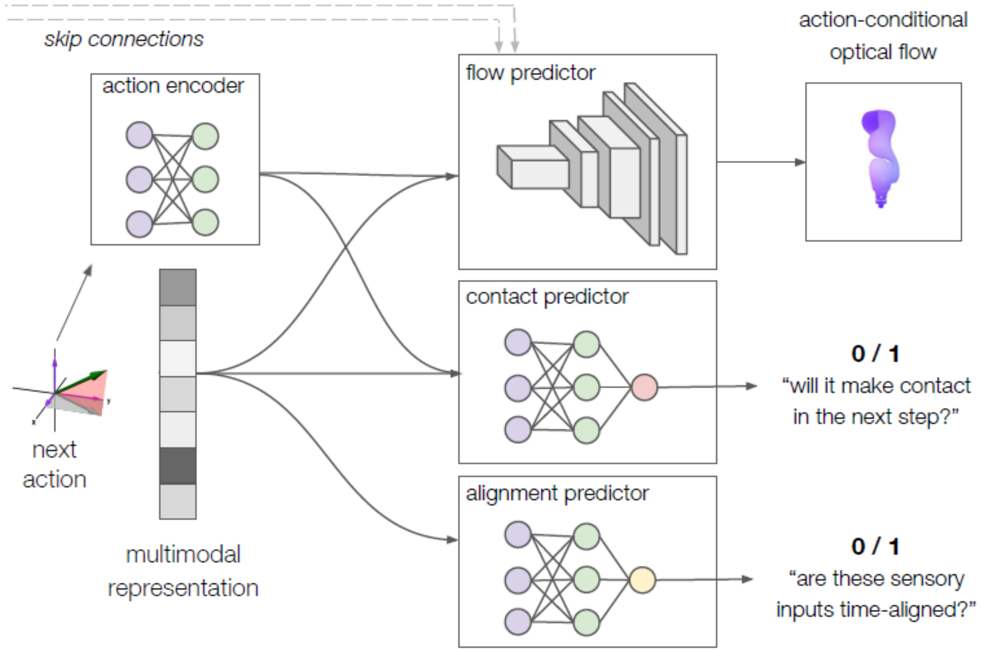

skip connections
action encoder
flow predictor
action-conditional optical flow
contact predictor
0 / 1
"will it make contact in the next step?"
next action
multimodal representation
alignment predictor
0 / 1
"are these sensory inputs time-aligned?"

Fig. 6: model for the representation learning [3]

- To learn representation, objective is needed
- 3 neural networks trained beforehand on data and are used for it
  - Optical flow predictor
  - Contact predictor
  - Alignment predictor (to check if the data from different modalities aligned over the time dimension)
- The labels are designed in such a way that they enable automated creation of dataset for training
  - Optical flow generated from kinematics based on proprioception
  - Heuristics (threshold, I guess) for contact
  - Misaligning the data-streams for the alignment predictor
- Contact and flow prediction are used, so that the multimodal representation will encode action-conditional relation
- Alignment predictor is used to help capturing the dependency between the modalities into the encoding

4.4 MODEL FOR REPRESENTATION LEARNING: SELF-SUPERVISED PREDICTIONS

- Action encoder:
  - Next action => 2-layer MLP => encoded next action
  - Next action is TCP motion
  - Forms input for the flow and contact predictor
- Flow predictor
  - Encoded next action + multimodal representation => 6-layer CNN decoder + upsampling with skip connections => 128x128 action conditional flow map
  - Endpoint error (EPE) loss averaged over all pixels

- Contract predictor
  - Encoded next action + multimodal representation => 2-layer MLP => "contact at the next step" 0/1
  - Cross entropy loss
- Alignment predictor
  - Randomly shifted in time/aligned multimodal data => 2-layer MLP => "time aligned or not" 0/1
  - Cross entropy loss

- Endpoint error (EPE)

  For a motion field H by W pixels,

$$\text{A-EPE} = \frac{\sum_{\mathbf{x}} \sqrt{(\hat{M}_{\boldsymbol{u}}[\mathbf{x}] - M_{\boldsymbol{u}}[\mathbf{x}])^2 + (\hat{M}_{\boldsymbol{v}}[\mathbf{x}] - M_{\boldsymbol{v}}[\mathbf{x}])^2}}{H \cdot W},$$

where $\hat{M} = (\hat{M}_{\boldsymbol{u}}, \hat{M}_{\boldsymbol{v}})$ and $M = (M_{\boldsymbol{u}}, M_{\boldsymbol{v}})$ denote the estimated and the ground truth motion fields, respectively.

- Cross entropy loss

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i),$$

where t_i is the binary truth label for class i,

p_i is the softmax probability

4.4 MODEL FOR REPRESENTATION LEARNING: SUMMARY

- Model is trained by stochastic gradient descent minimizing a sum of the three losses end-to-end
- Dataset made from rolled-out random and heuristic trajectories
- After the model is trained, we can use the encoder to get 128-d feature vector that compactly represents multimodal data
- This model is needed because designing encoder for the multimodal data by hand is infeasible
- Learned representation will be used as an input for policy learned by RL

4.5 POLICY LEARNING AND CONTROLLER DESIGN

- Again, designing controller by hand is infeasible, because it will be very task specific
- The idea is to enable self-supervised learning
- Therefore, RL will be used to learn the controller

4.5 POLICY LEARNING AND CONTROLLER DESIGN

- Policy Learning
  - Model-free reinforcement learning problem is used
  - Eliminates the need for an accurate dynamics model what may be difficult in this context
  - The trust-region policy optimization (TRPO) is used to optimize the policy
  - It imposes a bound of KL-divergence for each update, so the updated policy is not too far from the previous

- Policy network
  - 2-layer MLP
  - Input: 128-d multimodal representation
  - Output: 3D displacement dx of the robot end-effector
  - For the training efficiency, representation model is frozen

4.5 POLICY LEARNING AND CONTROLLER DESIGN

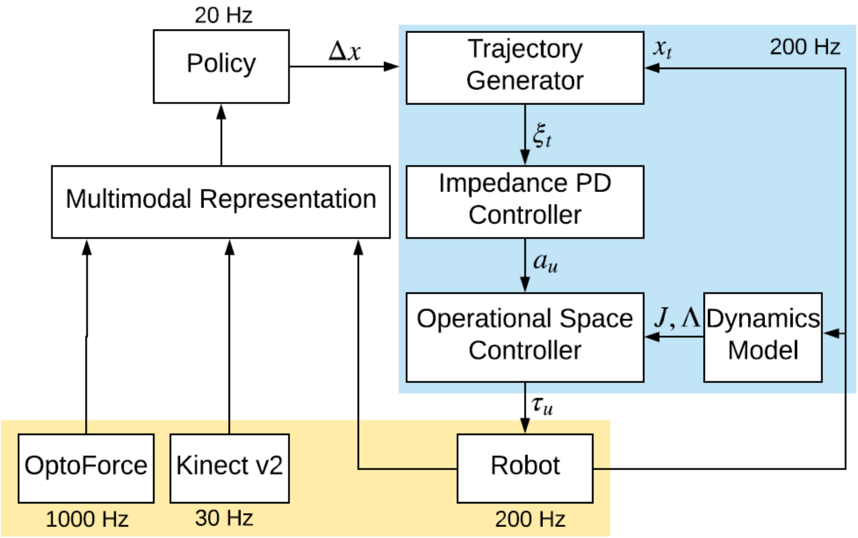


Fig. 7: model for the control and policy learning [3]

- Controller design
  - Input: dx form the policy 20 Hz
  - Output: direct torque commands τ_u 200 Hz
  - Policy uses cartesian space commands, so no complicated mapping between joint and cartesian spaces needs to be learned
  - Direct torque control allows for the compliance and increased safety

- Trajectory generator
  - Upsamples low-bandwidth output of the policy to the high-bandwidth needed for the controller from 20 Hz to 200 Hz
  - Calculates trajectory $\{x_k, v_k, a_k\}_{k=t}^{t+1}$, form current state and the give displacement dx
- Impedance PD controller
  - Calculates task space acceleration form the trajectory
  - $\mathbf{a}_u = \mathbf{a}_{\text{des}} - \mathbf{k}_{\text{p}}(\mathbf{x} - \mathbf{x}_{\text{des}}) - \mathbf{k}_{\text{v}}(\mathbf{v} - \mathbf{v}_{\text{des}})$
  - $k_{v,}k_p$ are manually tuned gained
- Dynamics model
  - Use dynamics and kinematics model of the robot
  - $\mathbf{F} = \Lambda \mathbf{a}_u$, where Λ inertial matrix in the end-effector
  - Map from **F** to τ_u with the Jacobian $\mathbf{q}$: $\tau_{\text{u}} = J^T(\mathbf{q})\mathbf{F}$.

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.6 EXPERIMENT DESIGN

Objectives of the experiments' setup:

1. What is the value of using all instead of a subset of modalities?
2. Is policy learning on the real robot practical with a learned representation?
3. Does the learned representation generalize over task variations and recover from perturbations?

4.6 EXPERIMENT DESIGN:
TASK SETUP

- The model is tested on the peg insertion task
- The input is haptic and visual information
- 5 types of pegs: round, square, triangular, semicircular, and hexagonal
- Clearance is 2mm

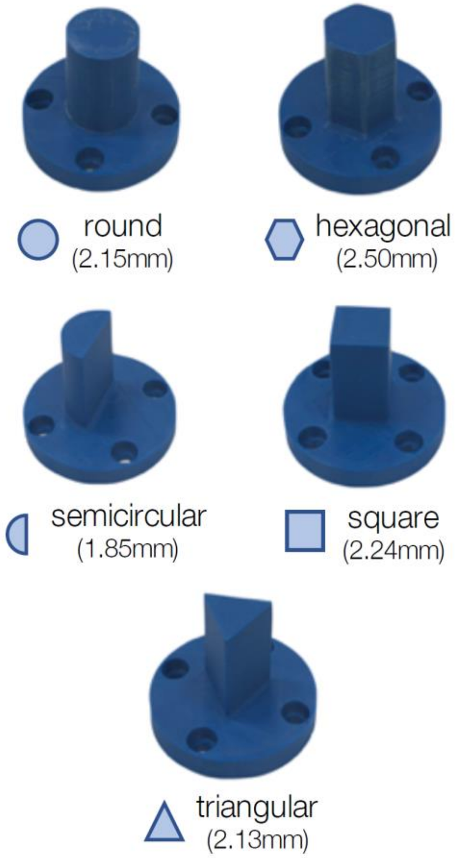


Fig. 8: Peg used for the peg insertion task[3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.6 EXPERIMENT DESIGN: REWARD DESIGN

$$r(\mathbf{s}) = \begin{cases} c_r - \frac{c_r}{2}(\tanh\lambda\|\mathbf{s}\| + \tanh\lambda\|\mathbf{s}_{xy}\|) & & \text{(reaching)} \\ 2 - c_a\|\mathbf{s}_{xy}\|_2 & \text{if } \|\mathbf{s}_{xy}\|_2 \leq \varepsilon_1 & \text{(alignment)} \\ 4 - 2(\frac{s_z}{h_d - \varepsilon_2}) & \text{if } s_z < 0 & \text{(insertion)} \\ 10 & \text{if } h_d - |s_z| \leq \varepsilon_2 & \text{(completion),} \end{cases}$$

- **s** - current peg position (planar or volumetric)
- λ – constant factor
- $(0, 0, -h_d)$- peg target position
- c – constant scaling factors

- The reward is built in such a way, to help the agent learn the insertion by consecutive subtasks, from reaching to insertion.
- The design of each subtasks' reward is simple: the closer peg position to the subtasks' goal is, the closer the reward is to the maximal of current subtask.

4.6 EXPERIMENT DESIGN: EVALUATION

- Sum of rewards achieved in episodes in % of the highest achievable reward
- Statistics on the progress of the task completion:
  1. completed insertion: the peg reaches bottom of the hole;
  2. inserted into hole: the peg goes into the hole but has not reached the bottom;
  3. touched the box: the peg only makes contact with the box;
  4. failed: the peg fails to reach the box.

4.6 EXPERIMENT DESIGN: ROBOT ENVIRONMENT SETUP

- Simulation and real hardware experiments
- Kuka LBR IIWA robot, 7-DoF, torque-controlled for both
- Modalities:
  - proprioception: end-effector pose, linear and angular velocity (from FK)
  - RGB camera: fixed camera pointed at the robot, downsampled to 128x128, Kinect v2 on the real hardware
  - Force-torque sensor: 6-axis forces and moments on x, y , z axes, OproForce sensor between the last joint and the peg on the hardware
- CHAI3D for rendering in the simulation https://www.chai3d.org
- SAI 2.0 for a real-time physics simulation to model the contact between the peg and the box https://github.com/manips-sai-org/sai2-simulation-release

4.6 EXPERIMENT DESIGN:
IMPLEMENTATION FOR REPRESENTATION LEARNING MODEL

- Collect a multimodal dataset of 100k states (90 to 120 minutes), generate the self-supervised annotations
- Use a random and heuristic policy for gathering the data (heuristic policy made to encourage the peg to make contact with the box)
- Policy is at 20 Hz
- Representation models are trained for 20 epochs on a Titan V GPU

4.6 EXPERIMENT DESIGN:
PEG INSERTION TASK

- Experiment consist of two parts:
  1. Simulation: to study the influence of the individual modalities on the policy learning
  2. Real robot experiment: apply full model for representation learning to train the policies for the insertion task

4.7 RESULTS: SIMULATION EXPERIMENTS

1. Learn the representation from different combinations of modalities.
2. Train TRPO policies to inserting a square peg.
   Randomize the box position and the arm's initial position for the episode initialization to improve generalizability and robustness. Policies trained with 1.2k episodes, 500 steps each.
3. 10 trials evaluated with the stochastic policy every 10 training steps. Mean and std of the episode rewards are reported.

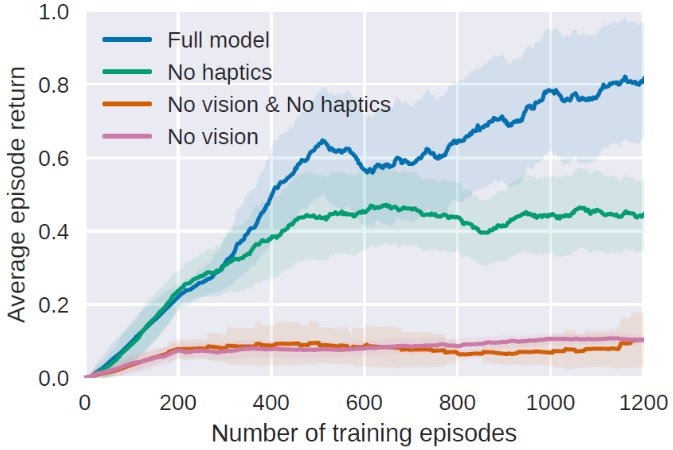


Fig.9: (a) Training curves of reinforcement learning [3]

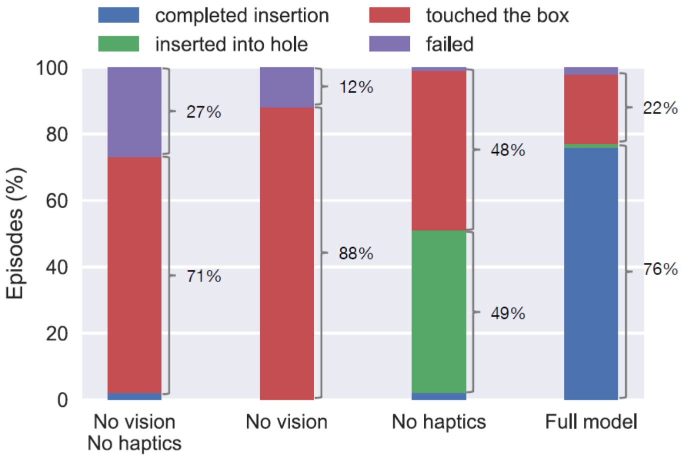


Fig.9: (b) Policy evaluation statistics [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.7 RESULTS:
SIMULATION EXPERIMENTS RESULTS

1. Unsurprisingly, the best representation includes all available modalities
2. It seems that vision plays important role for reaching and alignment, while the haptics is crucial to complete the insertion
3. Experiments prove the importance of all modalities for the consistent performance of the model

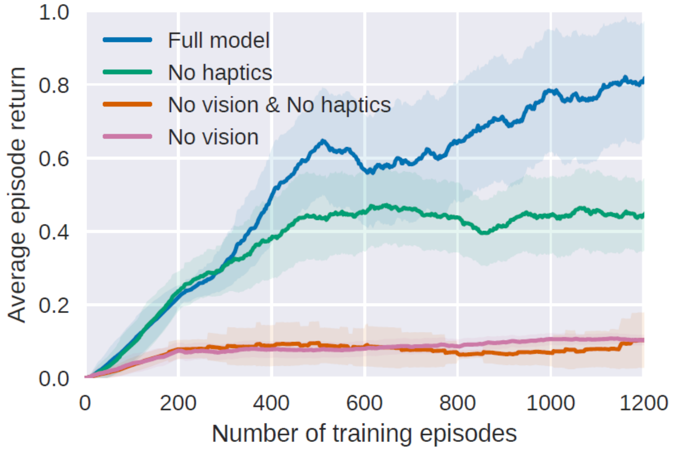


Fig.9: (a) Training curves of reinforcement learning [3]

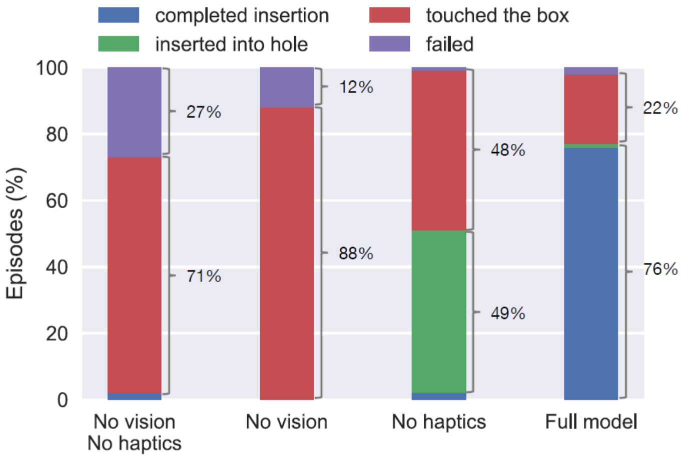
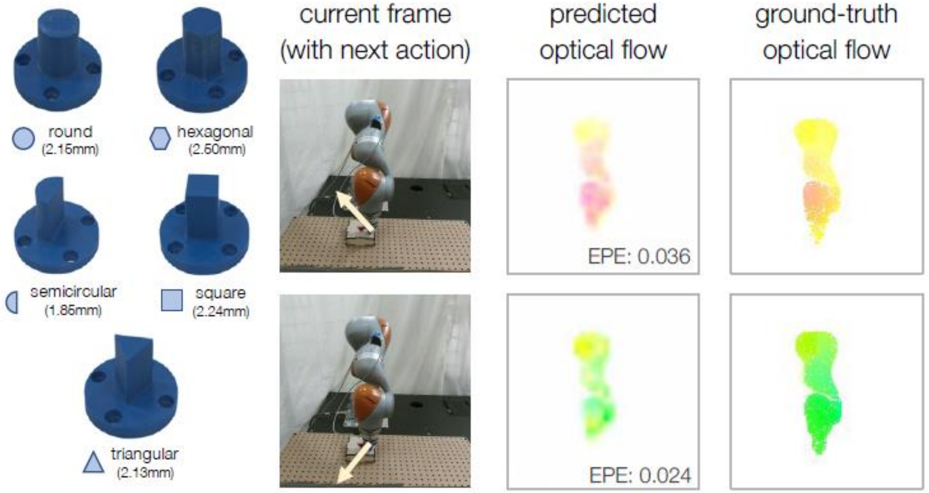


Fig.9: (b) Policy evaluation statistics [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.7 RESULTS:
REAL ROBOT EXPERIMENTS

- Learning policies for the peg with round, triangular, and semicircular pegs
- Full model for the representation learning
- Real hardware means new sources of uncertainties: sensor sync, sensing-control delays, complexity of the real-world physical interactions…
- Controller network is trained on the action conditional flows with the low endpoint error. To increase efficiently of the training, authors freeze weights of the trained model for the representation. This leaves to learn just 3% of the parameters from whole model.



(a) Peg variations (b) Optical flow prediction examples

Fig.10: Illustrations for the real robot experiments [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.7 RESULTS:
REAL ROBOT EXPERIMENTS

- Training:
  - TRPO policies are trained for 300 episodes
  - Each episode is 1000 steps, or ~5 hours of wall-clock time
  - Evaluate each policy for 100 episodes
- Results:
  - Achieved success rate is similar to the one in the simulation (how did they compare the results for the different peg shapes?☺)
  - Common learned behavior:
    1. Reach the box
    2. Search for the hole by sliding over the surface
    3. Align the peg with the hole
    4. Perform insertion.

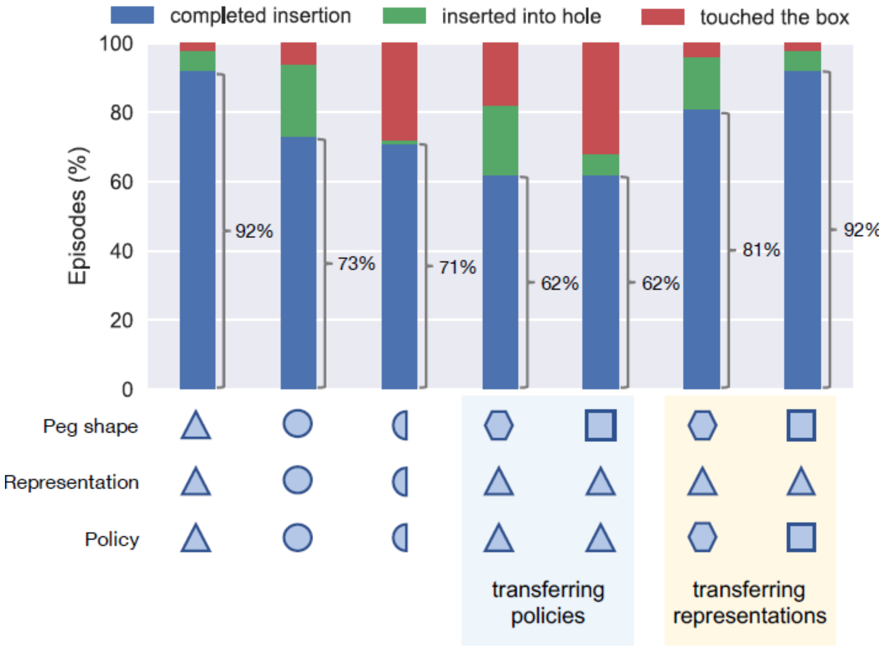


Fig.11: results of the real robot experiments [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.7 RESULTS: REAL ROBOT EXPERIMENTS

- Generalization of the learned policies and representations is studied
- This is done by testing learned policies and representation with the unseen hexagonal and square pegs
- Policy achieves 60% success rate for both pegs without training
- Further training of policy improves the success rate on 19% for the hexagonal peg and on 30% for the square peg
- This experiment showed good generalization of the learned representation

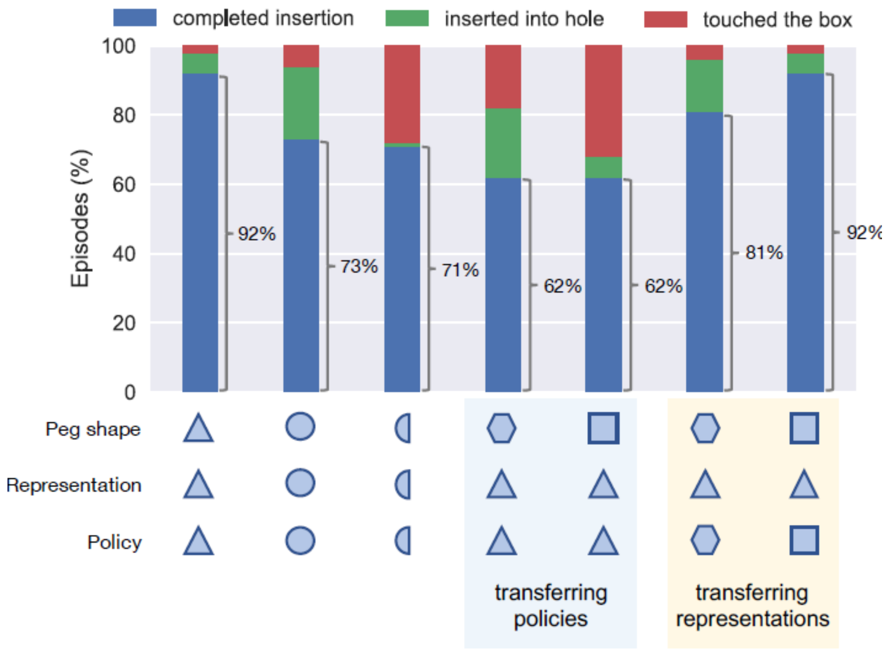


Fig.11: results of the real robot experiments [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.7 RESULTS:
REAL ROBOT EXPERIMENTS

- Bonus: the robustness of the policy to sensory noise in camera (by short occlusions) and external perturbations to the arm (pushing the robot arm during trajectory roll-out);
- The policy was able to recover from both the occlusions and the perturbations.

Video: Demo of all presented experiments [3]

[3] M. A. Lee et al., "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8943-8950, doi: 10.1109/ICRA.2019.8793485

4.8 CONCLUSIONS

- Key takeaways:
  - This work managed to bridge the reality gap
  - Use-case for the multi-modal data
  - Nice example how to design encoder for different modalities, that seems to generalize well for unseen tasks
  - Use-case for a self-supervised learning on the real hardware

QUESITONS?