Making Sense of Vision and Touch:

Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks

Presenter: Mao Ye

Sep 23, 2021

Motivation

Contact-rich manipulation tasks requires both haptic and visual feedback.

Goal: propose a **general/robust/generalizable** approach that is applicable to wide class of tasks. For example peg insertion with **different** shapes.

Why this is important:

- Real environment is with full of uncertainty and is unstructured. The robot must be robust.
- As objects can be different in real world, it’s better to use one robot for everything.

Key Challenge

Manual design of controller that combines modalities is very hard: seek for ML approach. However:

- Representation:
  - Haptic and visual feedback are quite different modals. How to do fusion?
- Learning:
  - Straightforward RL approach is sample inefficient.

General Idea

Decompose the learning into two stages:

- First stage: use self-supervision to learn good representation that fuses the multiple modals.
  - No need human labeling.
  - Easy to generate training data.
  - Not an MDP problem: easy to train.
- Fix the learned representations, conducting policy learning based on small network
  - Since number of trainable parameters is small, improved sample complexity.

Problem Setting

Goal: Learn a policy on a robot for performing contact-rich manipulation tasks

- Model the manipulation task as a finite-horizon, discounted Markov Decision Process (MDP).
- Maximize the expected discounted reward: $$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T-1} \gamma r(\mathbf{s}_t, \mathbf{a}_t)\right]$$
- Represent tha policy by neural network parameterized by θ. Input: state; output: action.

Related Work

Manipulation policies:

- Previous works often only reply on haptic feedback and force control but assume accurate state estimation (no visual input for state estimation) [1].
    - Usually one policy for one geometry [2]
    - or only limited a small range of shapes [3]

- [4] combines both vision and haptic but assuming known peg geometry.

Related Work

Reinforcement learning approaches:

- Seldom studies the complementary natural of vision and touch. Most of them do not combine the two modalities and do not work on full manipulation tasks [4,5,6,7].

- [8] uses multiple modalities but require a pre-specified manipulation graph and only works for single task.

Approach: Modality Encoders

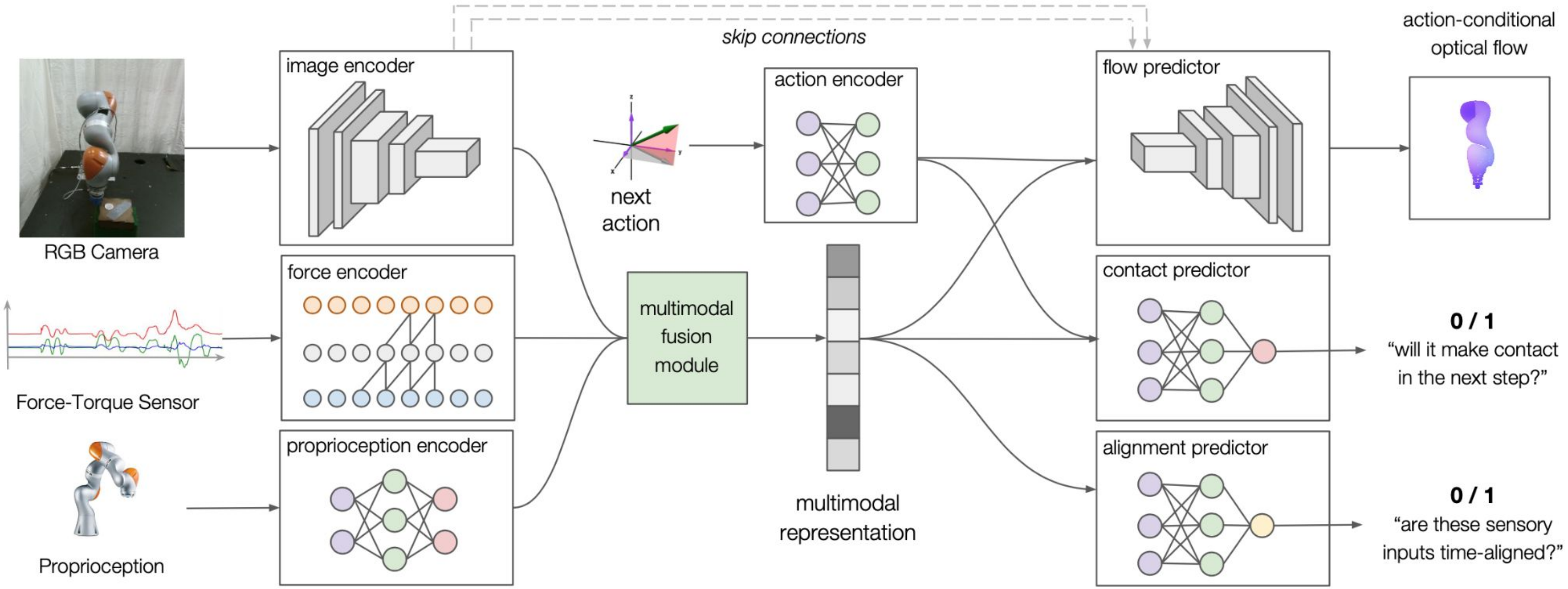


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Modality Encoders

6-layer Conv + MLP

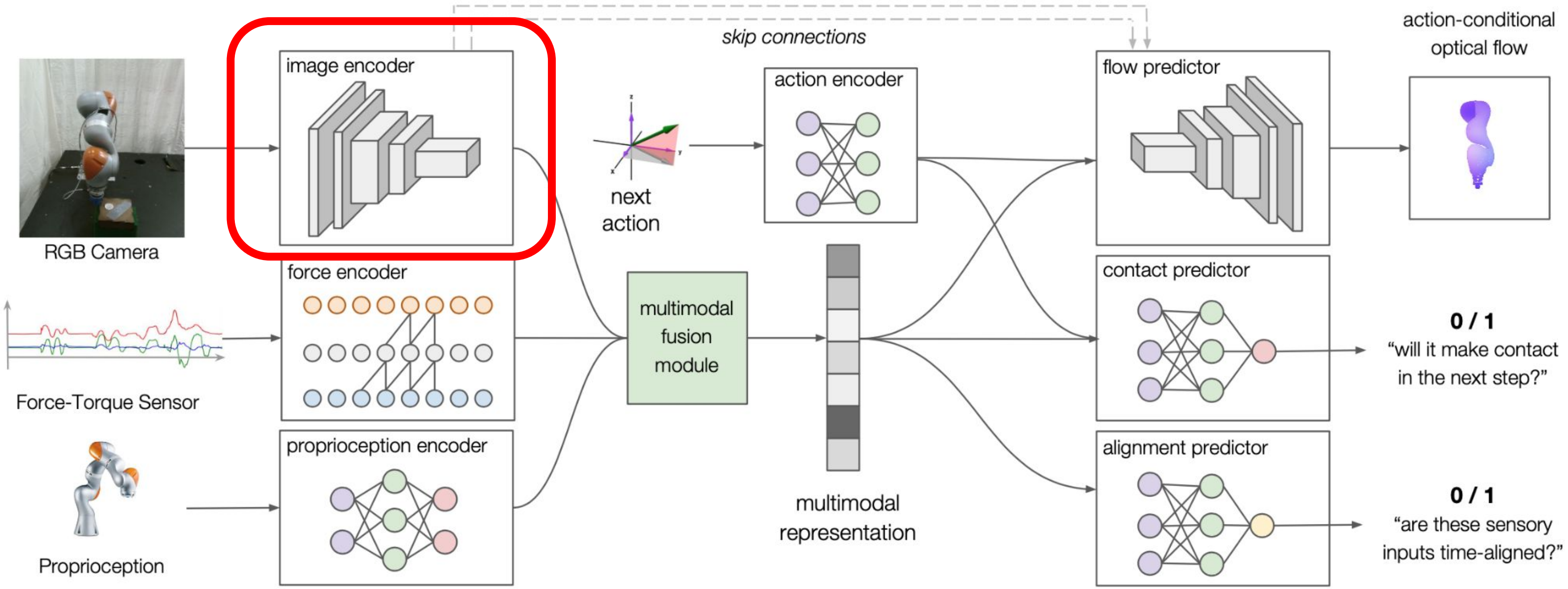


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Modality Encoders

Last 32 readings from 6-axis F/T sensor

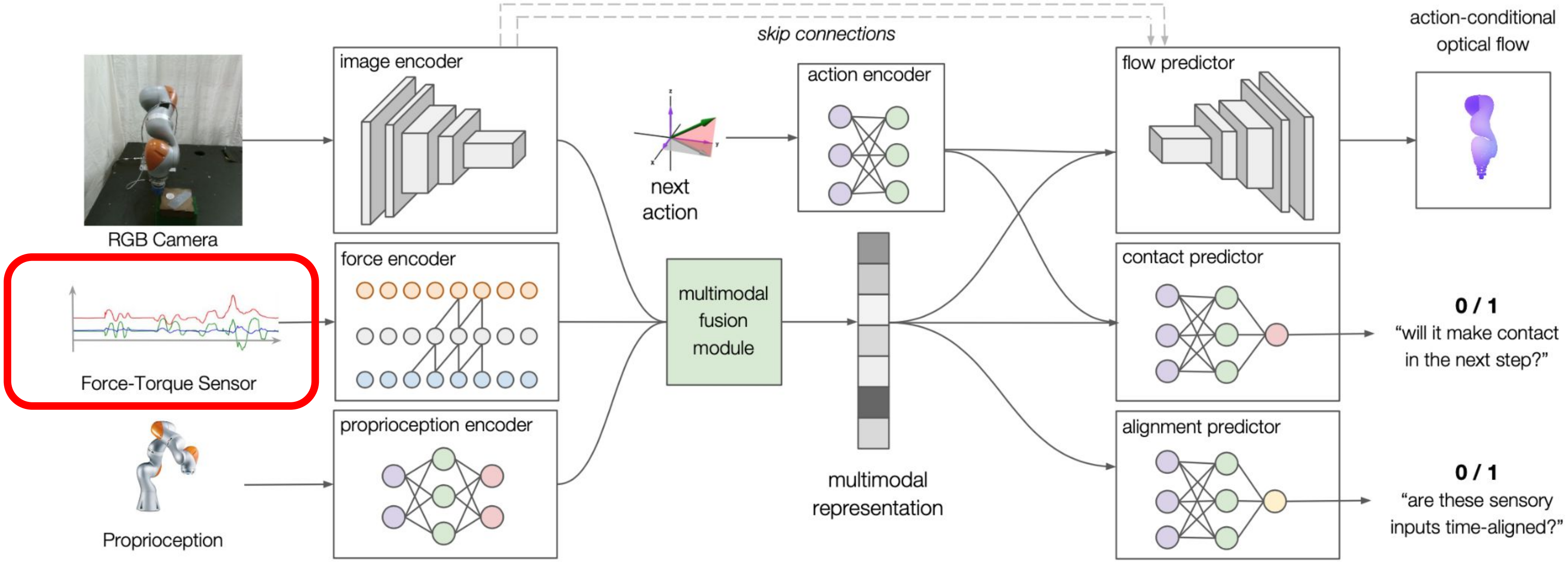


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

5-layer causal conv

Approach: Modality Encoders

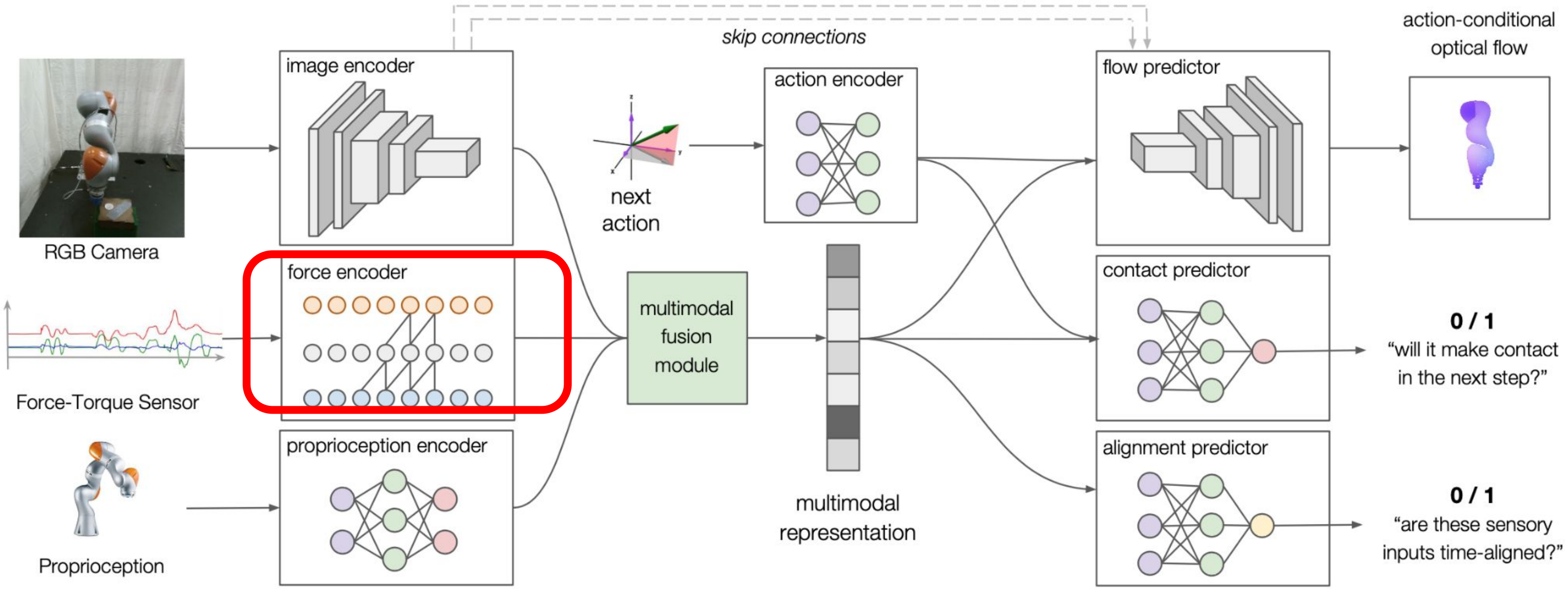


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Modality Encoders

Current position and velocity of the end-effector

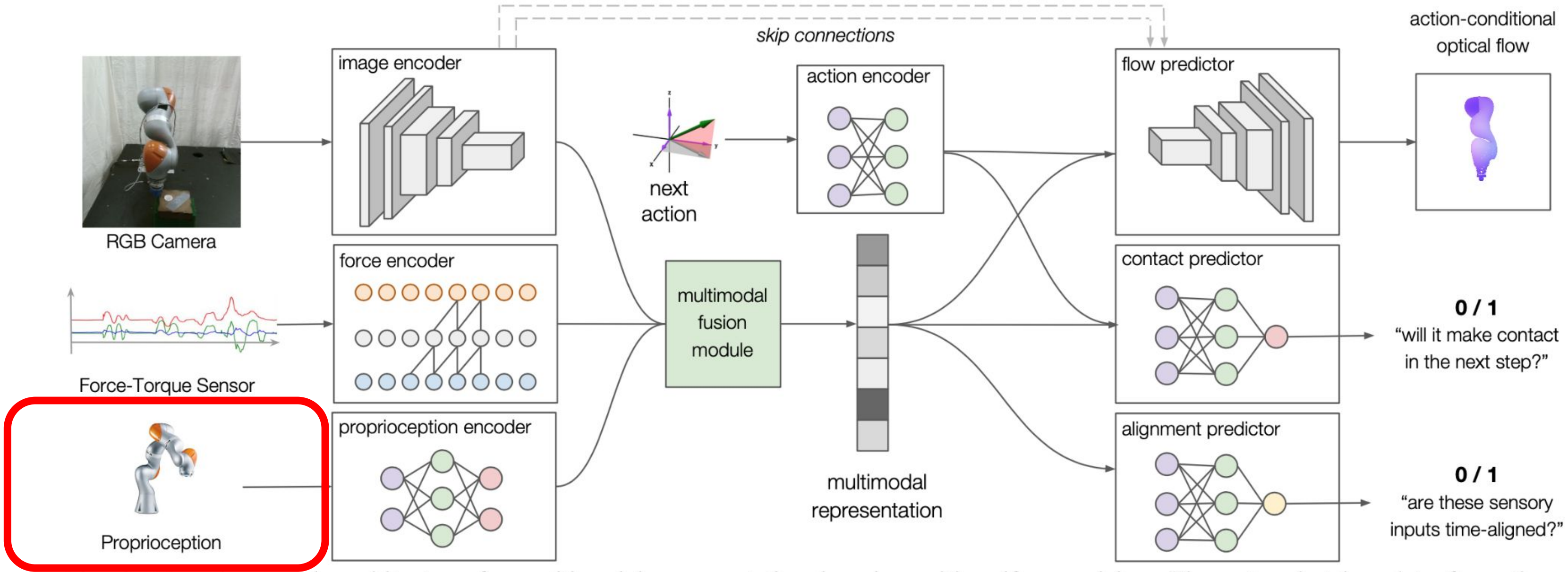


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

2-layer MLP

Approach: Modality Encoders

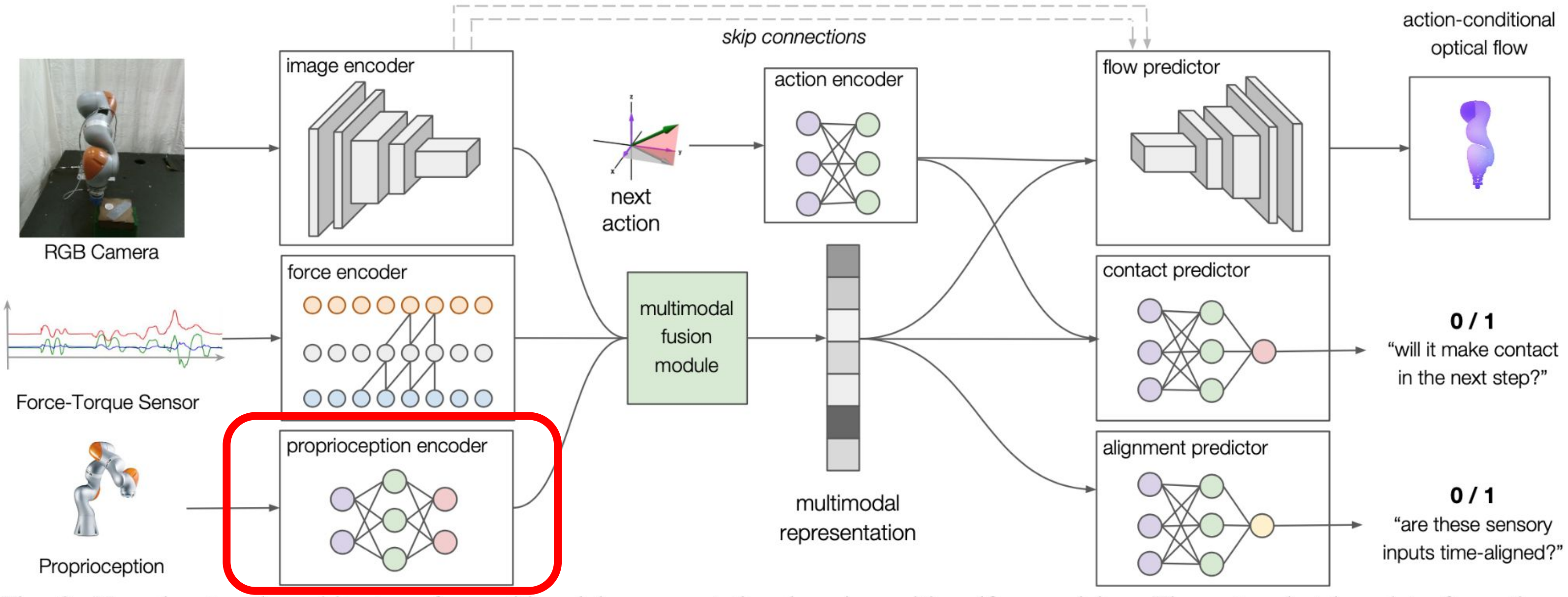


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Modality Encoders

Concat all the three vectors

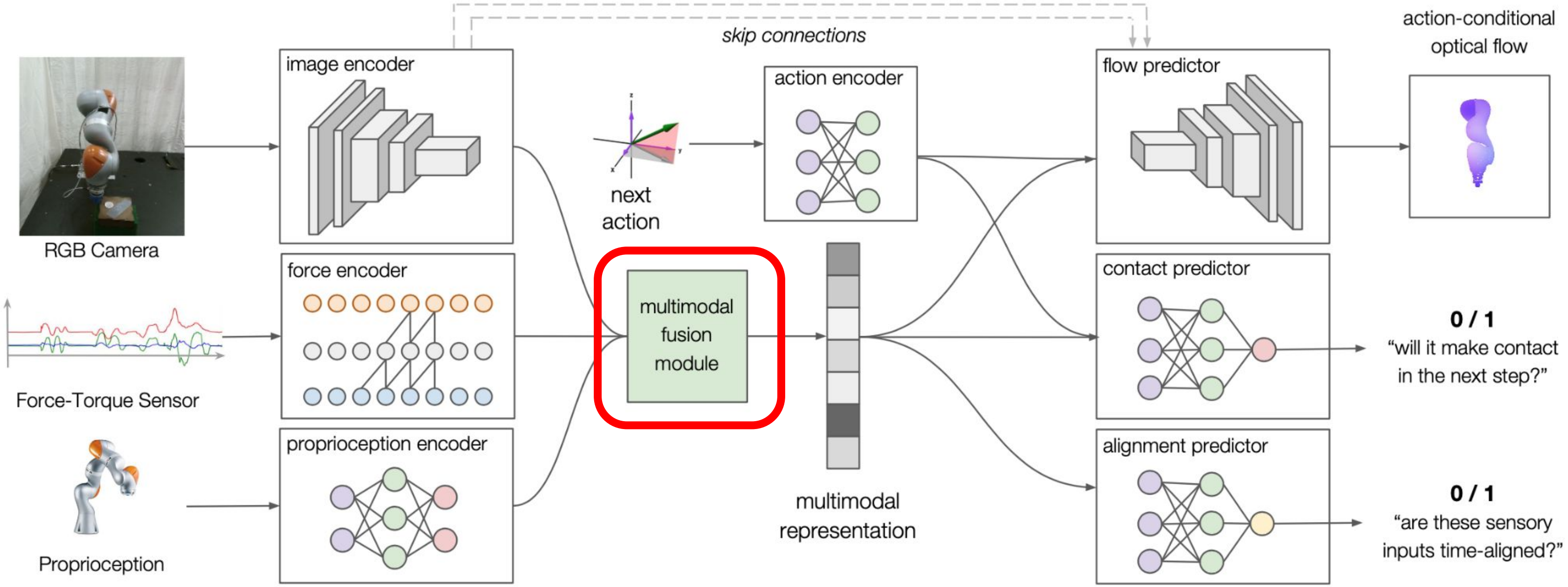


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Self-Supervised Tasks

Given action-conditional representation, we want to predict:

- Optical flow generated by the action
- Whether the end-effector will make contact with the environment in the next control cycle
- Whether two sensors streams are temporally aligned.
    - Previous literatures shows compelling evidence that the concurrency of different sensory streams aid perception and manipulation.

Approach: Self-Supervised Tasks

Given action-conditional representation, we want to predict:

- Optical flow generated by the action
  - Annotations are automatically generated given proprioception and known robot kinematics and geometry.
- Whether the end-effector will make contact with the environment in the next control cycle
  - Applying simple heuristics on the F/T readings.
- Whether two sensors streams are temporally aligned.
  - Not aligned streams are created manually (random shift)and thus naturally has the label.

Approach: Self-Supervised Training

endpoint error loss averaged over all pixels

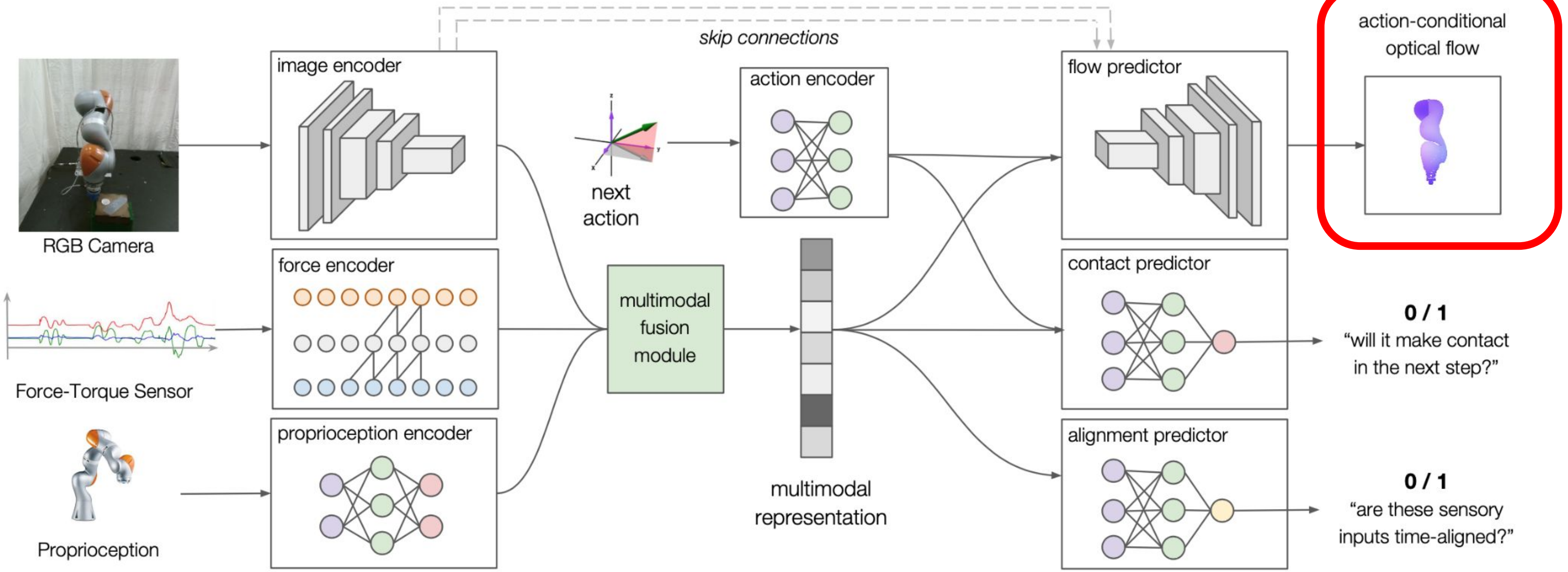


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Self-Supervised Training

Cross-entropy

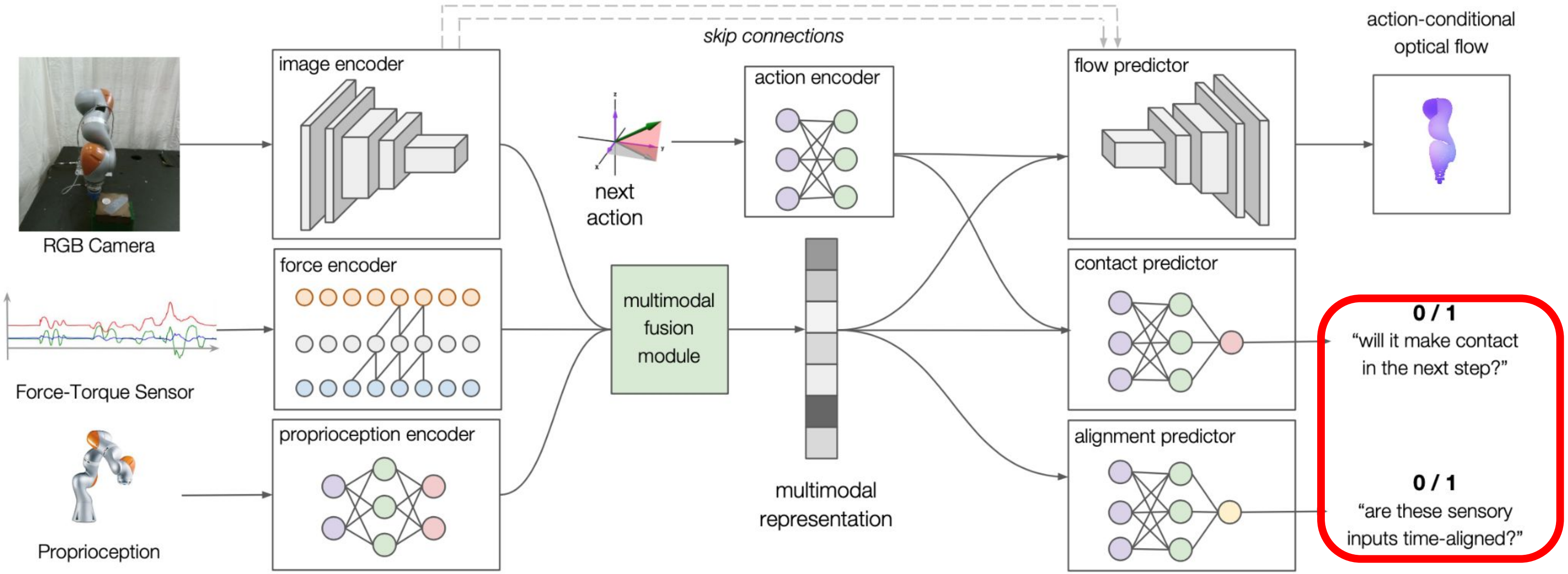


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation based on which controllers for contact-rich manipulation can be learned. This representation learning network is trained end-to-end through self-supervision.

Approach: Self-Supervised Training

- Training data
    - Obtain training data by applying heuristic algorithms for controlling the robot.

Approach: Policy Learning

Model-free reinforcement learning.

- Policy network: 2-layer MLP takes multimodal representation → 3D displacement of the robot effector.
    - Small network has good sample efficiency

- Training: trust-region policy optimization. Representation model parameters are frozen during training policy network.

Approach: Policy Learning

Reward Design:

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

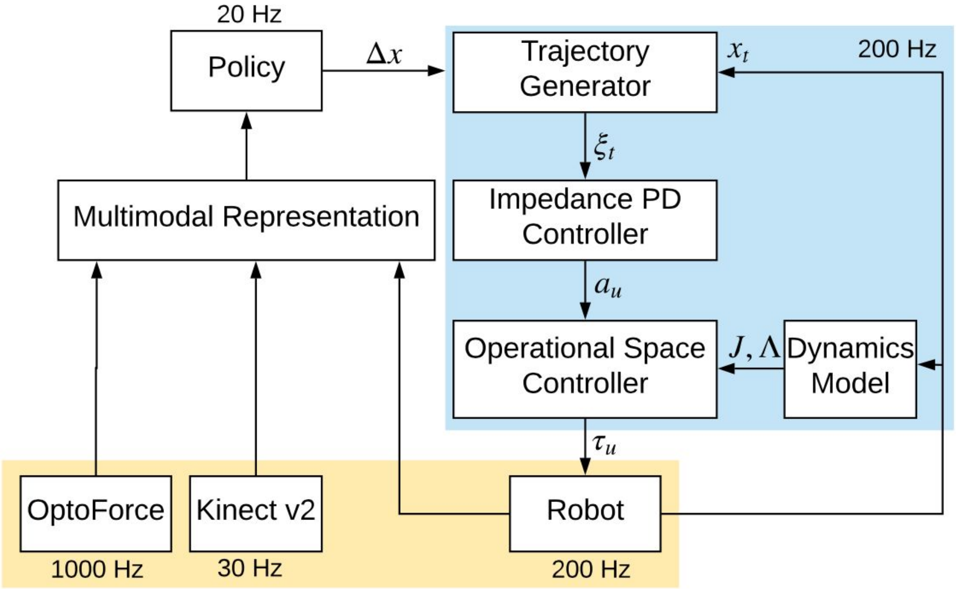
$$\mathbf{s} = (s_x, s_y, s_z) \qquad \mathbf{s}_{xy} = (s_x, s_y)$$

The target peg position is $(0, 0, -h_d)$

Approach: Controller Design

Input: end-effector displacement from the policy

Output: direct torque command to the robot.



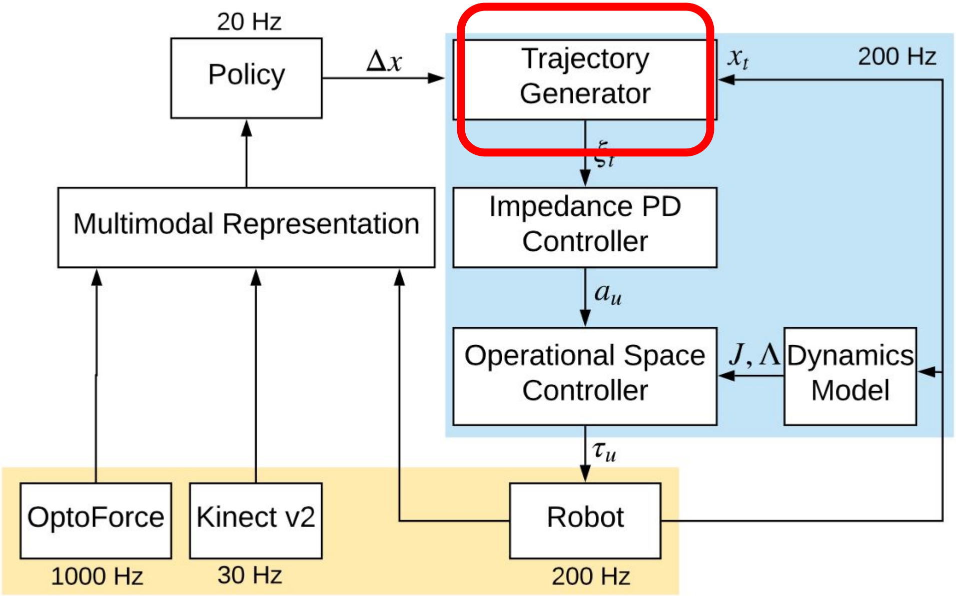
Approach: Controller Design

Input: end-effector displacement from the policy

Output: direct torque command to the robot.

Generate trajectory (position/velocity/acceleration) via Interpolating between start and end position

$$\xi_t = \{\mathbf{x}_k, \mathbf{v}_k, \mathbf{a}_k\}_{k=t}^{t+T}$$



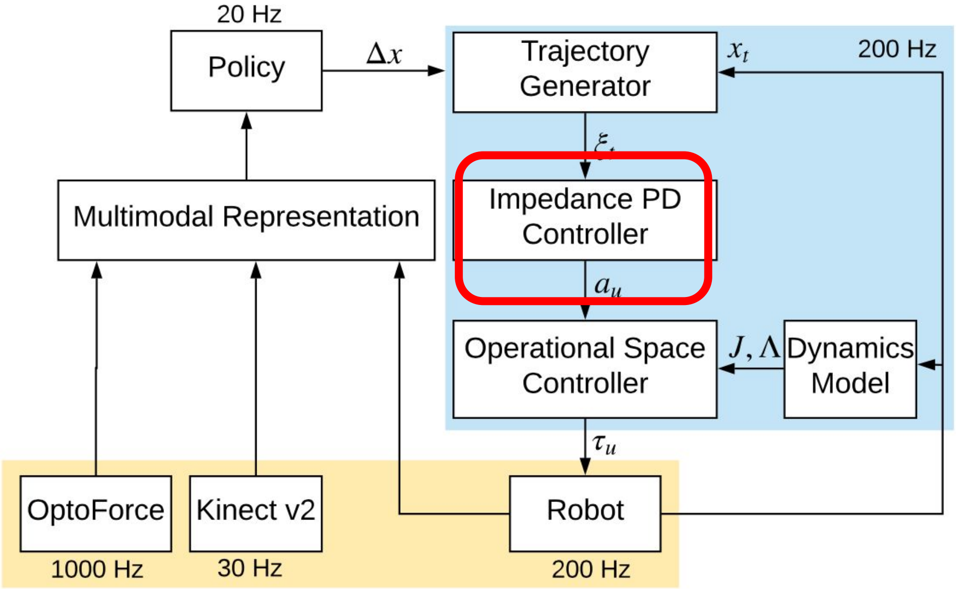
Approach: Controller Design

Input: end-effector displacement from the policy

Output: direct torque command to the robot.

PD impedance controller compute
task space acceleration commend

$$\mathbf{a}_u = \mathbf{a}_{\text{des}} - \mathbf{k}_{\text{p}}(\mathbf{x} - \mathbf{x}_{\text{des}}) - \mathbf{k}_{\text{v}}(\mathbf{v} - \mathbf{v}_{\text{des}})$$

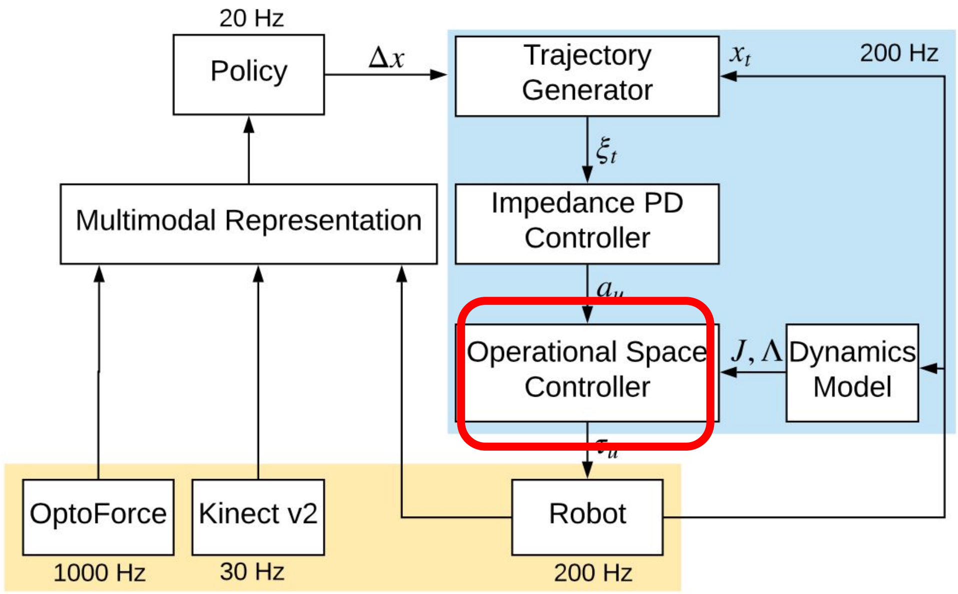


Approach: Controller Design

Input: end-effector displacement from the policy

Output: direct torque command to the robot.

Calculate the force needed



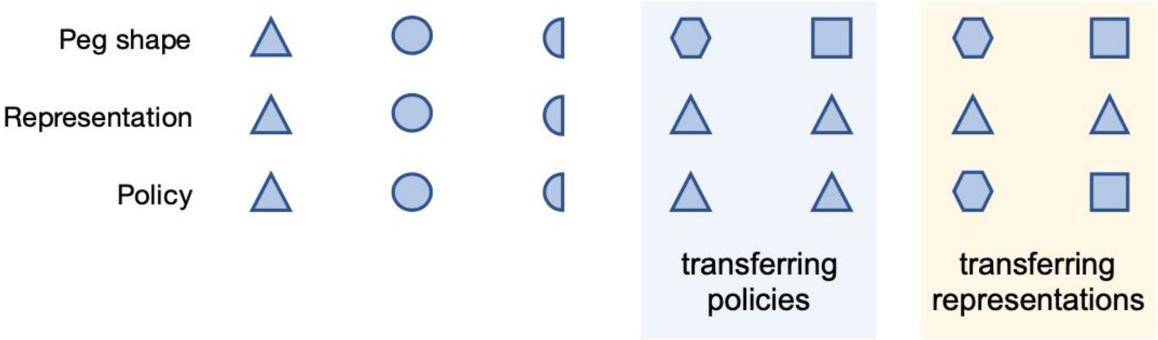
Experimental Setup

Key questions to answer:

- **What's the value of using all modalities instead of using part of them?**
- **Is policy learning on the real robot practical with a learned representation?**
- **Does the learned representation generalize over task variations and recover from perturbations?**

Experimental Setup

- Tasks
  - Peg insertion task with five different types of pegs and holes fabrication.



Experimental Setup

- Robot Environment Setup
    - Kuka LBR IIWA, a 7-DoF torque-controlled robot for both simulation and real robot experiment.

Experimental Setup

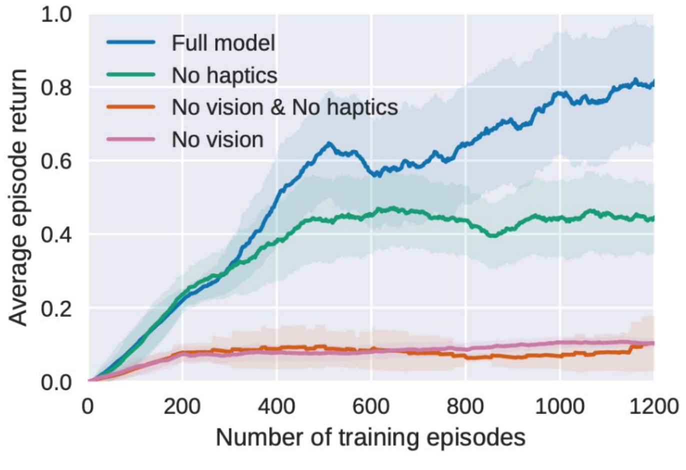
- **Evaluation Metrics**

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.

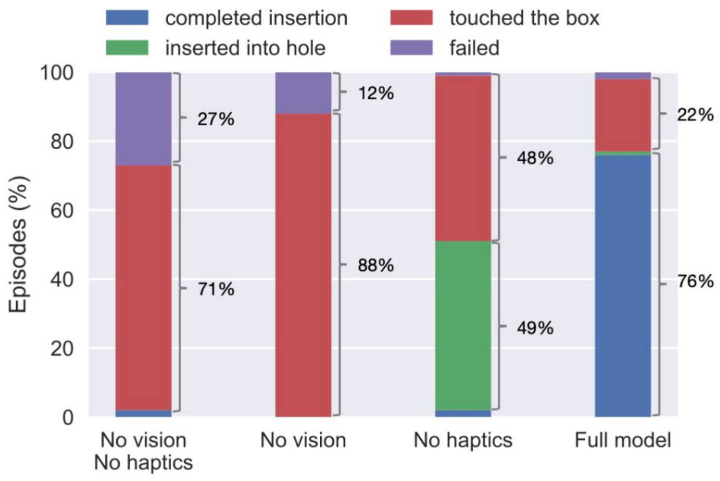
Experimental Results

What's the value of using all modalities instead of using part of them?

Design: ablation study on using different modalities. (Simulation)



(a) Training curves of reinforcement learning



(b) Policy evaluation statistics

Experimental Results

Is policy learning on the real robot practical with a learned representation?

Design: showing it works on real robot with reasonable training time.

TRPO policies are trained for 300 episodes: roughly 5 hours of wall-clock time

---- Pretty reasonable time

Works well according to the video in supplementary material.

https://sites.google.com/view/visionandtouch

Experimental Results

Is policy learning on the real robot practical with a learned representation?

Design: showing it works on real robot with reasonable training time.

TRPO policies are trained for 300 episodes: roughly 5 hours of wall-clock time

---- Pretty reasonable time

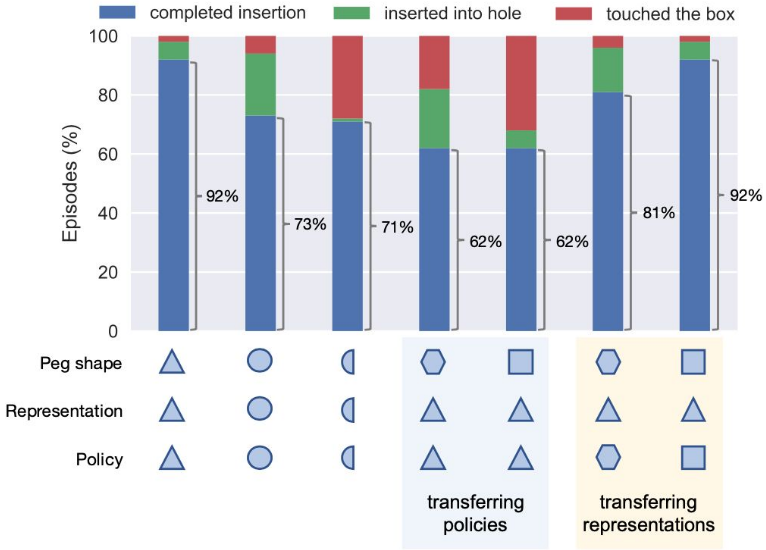
Works well according to the video in supplementary material.

https://sites.google.com/view/visionandtouch

Experimental Results

Does the learned representation generalize over task variations and recover from perturbations?

Design: Transfer learning and showing robust to external perturbation (see video).



* * Representations are easier to transfered

Discussion of Results

The goodness:

- Experiment results gives good support for the three main questions that this paper want to answer.
- The design is very suitable for answering the question.
- The results are very solid!

Discussion of Results

The weakness:

- Evidence for transfer learning seems not that strong. Only limited pairs are provided. And all the results uses triangle as source.
- The representation learning pipeline is not discussed in the paper? We train the representation using the simulation or real robot?
- Seems the learned algorithm is only able to plug a certain shape of peg. Is it possible to train the robot so that it can handle multiple shapes of peg? Would such training gives a even more robust solution with better generalization ability?
- Sample complexity is not studied, while this is one motivation of the paper. What will happen if we increase/decrease the sample for representation/robot learning? How the two-stage learning benefits over the end-to-end learning?

Future Work

- How to train the network so that it is able to handle many geometries.
  - A single network trained with multiple geometrics?
  - A multi-task network that first detect the shape and then choose a subnet?

- What task (geometry) would be the one that gives the best generalization ability?
  - Parameterize the task and use meta-learning?

- What is the auxiliary task to improve the performance?
  - 2D detection so that the model is more aware of the location of the hole? Or use the 2D detection to localize the hole first to reduce the time for plugging?

Extended Readings

Many of the follow up works focus on building a more robust robot:

- **Dealing with uncertain holes:** https://arxiv.org/pdf/1902.09157.pdf
- Studying the robustness of multi-modal fusion. https://www.mdpi.com/2079-9292/9/7/1152, https://www.merl.com/publications/docs/TR2020-110.pdf
- Scalability: how to train so that the model is able to learn to insert with many different shapes https://arxiv.org/pdf/2104.14223.pdf

Summary

Contributions:

- Whether/How to fuse the vision and haptic to enhance the peg plug performance.
- Use self-supervision and two-stage training to reduce the sample complexity for policy learning
- Showing the solution practical in real world robot.

Limitation:

- Generality of the functionality can be improved? More robust/ solve more task with one algorithm?

Key insight:

- Self-supervision is able to learn good representation and effectively reduce sample complexity.
- Multi-modal fusion is very useful

Reference

[1] D. E. Whitney, “Historical perspective and state of the art in robot force control”, *Int. J. Rob. Res.*, vol. 6, no. 1, pp. 3–14, Mar. 1987.

[2] D. E. Whitney, “Quasi-Static Assembly of Compliantly Supported Rigid Parts”, *Journal of Dynamic Systems, Measurement, and Con- trol*, vol. 104, no. 1, pp. 65–77, 1982.

[3] M. E. Caine, T. Lozano-Perez, and W. P. Seering, “Assembly strategies for chamferless parts”, in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, 472–477 vol.1.

[4] H. van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters, “Stable reinforcement learning with autoencoders for tactile and vi- sual data”, in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 3928–3934.

[5] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learn- ing force control policies for compliant manipulation”, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4639–4644.

[6] J. Sung, J. K. Salisbury, and A. Saxena, “Learning to represent haptic feedback for partially-observable tasks”, in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, pp. 2802–2809.

[7] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, “Learning robot in-hand manipulation with tactile features”, in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, IEEE, 2015, pp. 121–127.

[8] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, “Adaptation of manipulation skills in physical contact with the environment to reference force profiles”, *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015.

[9] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. Mc- Grew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, “Learning dexterous in-hand manipulation”, *ArXiv preprint arXiv:1808.00177*, 2018.

[10] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, “Using simulation and domain adaptation to improve efficiency of deep robotic grasping”, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 4243–4250.

[11] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim- to-real transfer of robotic control with dynamics randomization”, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–8.