Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks

Michelle A. Lee*, Yuke Zhu*, Krishnan Srinivasan, Parth Shah,
Silvio Savarese, Li Fei-Fei, Animesh Garg, Jeannette Bohg

***Abstract*— Contact-rich manipulation tasks in unstructured environments often require both haptic and visual feedback. However, it is non-trivial to manually design a robot controller that combines modalities with very different characteristics. While deep reinforcement learning has shown success in learning control policies for high-dimensional inputs, these algorithms are generally intractable to deploy on real robots due to sample complexity. We use self-supervision to learn a compact and multimodal representation of our sensory inputs, which can then be used to improve the sample efficiency of our policy learning. We evaluate our method on a peg insertion task, generalizing over different geometry, configurations, and clearances, while being robust to external perturbations. Results for simulated and real robot experiments are presented.**

I. Introduction

Even in routine tasks such as putting a car key in the ignition, humans effortlessly combine our senses of vision and touch to complete the task. Visual feedback provides information about semantic and geometric object properties for accurate reaching or grasp pre-shaping. Haptic feedback provides information about the current contact conditions between object and environment for accurate localization and control even under occlusions. These two feedback modalities are complementary and concurrent during contact-rich manipulation [6]. Yet, there are few algorithms that endow robots with an ability similar to humans. While the utility of multimodal data has been shown in robotics frequently [5, 38, 41, 46], the proposed manipulation strategies are often task-specific. On the other hand, while learning-based methods do not require manual task specification, the majority of learned manipulation policies close the control loop around vision only [12, 17, 28, 50].

In this work, we equip a robot with a policy that leverages multimodal feedback from vision and touch. This policy is learned through self-supervision and generalizes over variations of the same contact-rich manipulation task in geometry, configurations, and clearances. It is also robust to external perturbations. Our approach starts with using neural networks to learn a shared representation of haptic and visual sensory data, two modalities with very different dimensions, frequencies, and characteristics. Using a self-supervised learning objective, this network is trained to predict optical flow, whether contact will be made in the next control cycle, and concurrency of visual and haptic data. The training is action-conditional to encourage the state representation to encode action-related information. The resulting compact representation of the high-dimensional and heterogeneous data is the input to a policy for contact-rich manipulation tasks using deep reinforcement learning. The proposed decoupling of state estimation and control achieves practical sample efficiency for learning both representation and policy on a real robot. Our primary contributions are three-fold:

1) A model for multimodal representation learning from which a contact-rich manipulation policy can be learned.
2) Demonstration of insertion tasks that effectively utilize both haptic and visual feedback for hole search, peg alignment, and insertion. Ablative studies compare the effects of each modality on task performance.
3) Evaluation of generalization to tasks with different peg geometry and of robustness to perturbation and sensor noise.

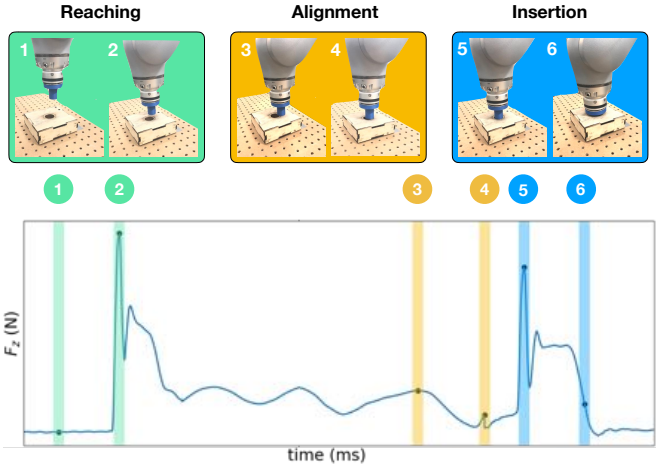


Fig. 1: Force sensor readings in the z-axis (height) and visual observations are shown with corresponding stages of the peg insertion task. When reaching for the box, the force reading transitions from (1) the arm being in free space to (2) being in contact with the box. While aligning the peg, the forces capture the dynamics of contact as the peg slides on the box surface (3, 4). Finally, in the insertion stage, the forces peak as the robot attempts to insert the peg at the edge of the hole (5), and decreases when the peg slides into the hole (6).

II. Related Work and Background

A. Contact-Rich Manipulation

Contact-rich tasks, such as peg insertion, block packing, and edge following, have been studied for decades due to their relevance in manufacturing. Manipulation policies often rely entirely on haptic feedback and force control,

*Authors have contributed equally and names are in alphabetical order.
All authors are with the Department of Computer Science, Stanford University. [mishlee, yukez, krshna, pshah9, ssilvio, feifeili, animeshg, bohg]@stanford.edu]. Animesh Garg is also at Nvidia Research.

and assume sufficiently accurate state estimation [48]. They typically generalize over certain task variations, for instance, peg-in-chamfered-hole insertion policies that work independently of peg diameter [47]. However, entirely new policies are required for new geometries. For chamferless holes, manually defining a small set of viable contact configurations has been successful [9] but cannot accommodate the vast range of real-world variations. [41] combines visual and haptic data for inserting two planar pegs with more complex cross sections, but assumes known peg geometry.

Reinforcement learning approaches have recently been proposed to address unknown variations in geometry and configuration for manipulation. [28, 50] trained neural network policies using RGB images and proprioceptive feedback. Their approach works well in a wide range of tasks. However, the large object clearances compared to automation tasks may explain the sufficiency of RGB data. A series of learning-based approaches have relied on haptic feedback for manipulation. Many of them are concerned with the problem of estimating the stability of a grasp before lifting an object [4, 11], potentially even suggesting a regrasp [43]. Only a few approaches learn entire manipulation policies through reinforcement only given haptic feedback [21, 22, 44]. While [22] relies on raw force-torque feedback, [21, 44] learn a low-dimensional representation of high-dimensional tactile data before learning a policy. Even fewer approaches exploit the complementary nature of vision and touch. Some of them extend their previous work on grasp stability estimation [3, 10]. Others perform full manipulation tasks based on multiple input modalities [23] but require a pre-specified manipulation graph and demonstrate only on a single task.

B. Multimodal Representation Learning

The complementary nature of heterogeneous sensor modalities has previously been explored for inference and decision making. The diverse set of modalities includes vision, range, audio, haptic and proprioceptive data as well as language. This heterogeneous data makes the application of hand-designed features and sensor fusion extremely challenging. That is why learning-based methods have been on the forefront. [3, 10, 19, 40] are examples of fusing visual and haptic data for grasp stability assessment, manipulation, material recognition, or object categorization. [30, 44] fuse vision and range sensing and [44] adds language labels. While many of these multimodal approaches are trained through a classification objective [3, 10, 19, 49], in this paper we are interested in multimodal representation learning for control. A popular representation learning objective is reconstruction of the raw sensory input [8, 21, 27, 49]. This unsupervised objective benefits learning stability and speed, but it is also data intensive and prone to overfitting [8]. When learning for control, action-conditional predictive representations are beneficial as they encourage the state representations to capture action-relevant information [27]. Studies attempted to predict full images when pushing objects with benign success [1, 2, 32]. In these cases either the underlying dynamics is deterministic [32], or the control runs at a low frequency [17]. In contrast, we operate with haptic feedback at 1kHz and send Cartesian control commands at 20Hz. We use an action-conditional surrogate objective for predicting optical flow and contact events with self-supervision.

There is compelling evidence that the interdependence and concurrency of different sensory streams aid perception and manipulation [7, 15, 25]. However, few studies have explicitly exploited this concurrency in representation learning. Examples include [42] for visual prediction tasks and [31, 34] for audio-visual coupling. Following [34], we propose a self-supervised objective to fuse visual and haptic data.

III. Problem Statement and Method Overview

Our goal is to learn a policy on a robot for performing contact-rich manipulation tasks. We want to evaluate the value of combining multisensory information and the ability to transfer multimodal representations across tasks. For sample efficiency, we first learn a neural network-based feature representation of the multisensory data. The resulting compact feature vector serves as input to a policy that is learned through reinforcement learning.

We phrase the problem as a finite-horizon discounted Markov Decision Process (MDP) $\mathcal{M}$, with a state space $\mathcal{S}$, an action space $\mathcal{A}$, state transition dynamics $\mathcal{T}: \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, an initial state distribution ρ_0, a reward function $r: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, horizon T, and discount factor $\gamma \in (0, 1]$. We are interested in maximizing the expected discounted reward

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

to determine the optimal stochastic policy $\pi: \mathcal{S} \to \mathbb{P}(\mathcal{A})$. We represent the policy by a neural network with parameters θ_π that are learned as described in Sec. V. $\mathcal{S}$ is defined by the low-dimensional representation learned from high-dimensional visual and haptic sensory data. This representation is a neural network parameterized by θ_s and is trained as described in Sec. IV. $\mathcal{A}$ is defined over continuously-valued, 3D displacements $\Delta\mathbf{x}$ in Cartesian space. The controller design is detailed in Sec. V.

IV. Multi-Modal Representation Model

Deep networks are a powerful tool to learn representations from high-dimensional data [26] but require a substantial amount of training data. Here, we address the challenge of seeking sources of supervision that do not rely on laborious human annotation. We design a set of predictive tasks that are suitable for learning visual and haptic representations for contact-rich manipulation tasks, where supervision can be obtained via automatic procedures rather than manual labeling. Figure 2 visualizes our representation learning model.

A. Modality Encoders

Our model encodes three types of sensory data available to the robot: RGB images from a fixed camera, haptic feedback from a wrist-mounted force-torque (F/T) sensor, and proprioceptive data from the joint encoders of the robot

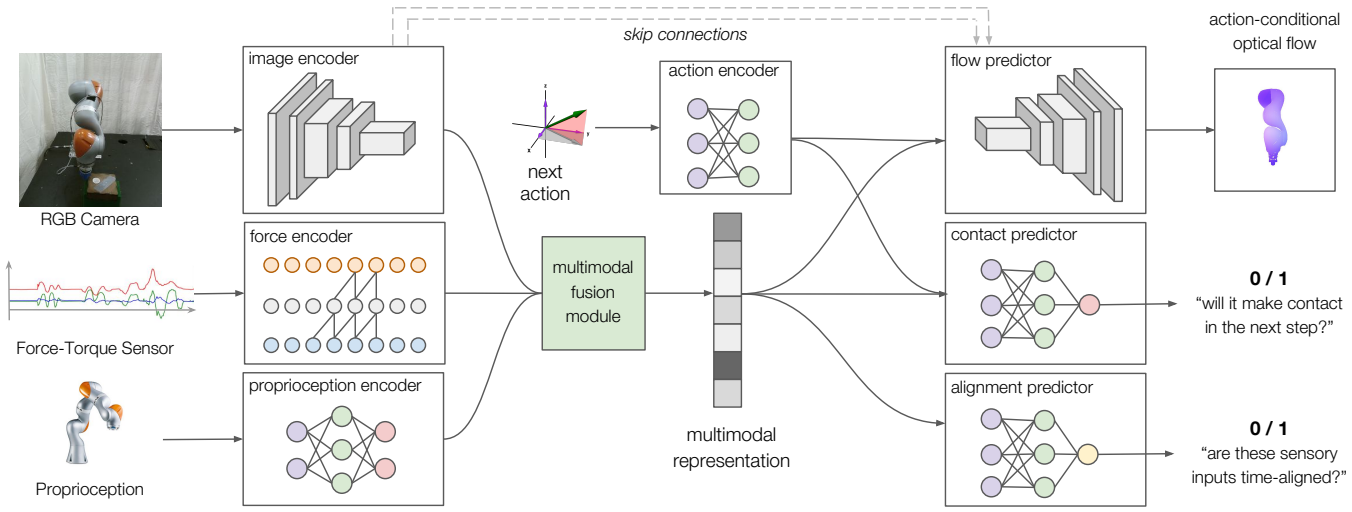


Fig. 2: Neural network architecture for multimodal representation learning with self-supervision. The network takes data from three different sensors as input: RGB images, F/T readings over a 32ms window, and end-effector position and velocity. It encodes and fuses this data into a multimodal representation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

arm. The heterogeneous nature of this data requires domain-specific encoders to capture the unique characteristics of each modality. For visual feedback, we use a 6-layer *convolutional neural network* (CNN) similar to FlowNet [18] to encode $128 \times 128 \times 3$ RGB images. We add a fully-connected layer to transform the final activation maps into a 128-d feature vector. For haptic feedback, we take the last 32 readings from the six-axis F/T sensor as a 32×6 time series and perform 5-layer causal convolutions [33] with stride 2 to transform the force readings into a 64-d feature vector. For proprioception, we encode the current positions and velocities of the end-effector with a 2-layer *multilayer perceptron* (MLP) to produce a 32-d feature vector. The resulting three feature vectors are concatenated into one vector and passed through the multimodal fusion module (2-layer MLP) to produce the final 128-d multimodal representation.

B. Self-Supervised Predictions

The modality encoders have nearly half a million learnable parameters and require a large amount of labeled training data. To avoid manual annotation, we design training objectives for which labels can be automatically generated through self-supervision. Furthermore, representations for control should encode the action-related information. To achieve this, we design two action-conditional representation learning objectives. Given the next robot action and the compact representation of the current sensory data, the model has to predict (i) the optical flow generated by the action and (ii) whether the end-effector will make contact with the environment in the next control cycle. Ground-truth optical flow annotations are automatically generated given proprioception and known robot kinematics and geometry [18, 20]. Ground-truth annotations of binary contact states are generated by applying simple heuristics on the F/T readings.

The next action, i.e. the end-effector motion, is encoded by a 2-layer MLP. Together with the multimodal representation it forms the input to the flow and contact predictor. The flow predictor uses a 6-layer convolutional decoder with upsampling to produce a flow map of size $128 \times 128 \times 2$. Following [18], we use skip connections. The contact predictor is a 2-layer MLP and performs binary classification.

As discussed in Sec. II-B, there is concurrency between the different sensory streams leading to correlations and redundancy, e.g., seeing the peg, touching the box, and feeling the force. We exploit this by introducing a third representation learning objective that predicts whether two sensor streams are temporally aligned [34]. During training, we sample a mix of time-aligned multimodal data and randomly shifted ones. The alignment predictor (a 2-layer MLP) takes the low-dimensional representation as input and performs binary classification of whether the input was aligned or not.

We train the action-conditional optical flow with the endpoint error (EPE) loss averaged over all pixels [18], and both the contact prediction and the alignment prediction with cross-entropy loss. During training, we minimize a sum of the three losses end-to-end with stochastic gradient descent on a dataset of rolled-out trajectories. Once trained, this network produces a 128-d feature vector that compactly represents multimodal data. This vector is taken as input to the manipulation policy learned via reinforcement learning.

V. Policy Learning and Controller Design

Our final goal is to equip a robot with a policy for performing contact-rich manipulation tasks that leverage multimodal feedback. Though it is possible to engineer controllers for specific instances of these tasks [41, 48], this effort is difficult to scale up due to the large variability of real-world tasks. Therefore, it is desirable to enable a robot to supervise itself where the learning process is applicable to a broad range of tasks. Given its recent successes in continuous control [29, 39], deep reinforcement learning is regarded as a natural choice for learning policies that transform high-dimensional features to control commands.

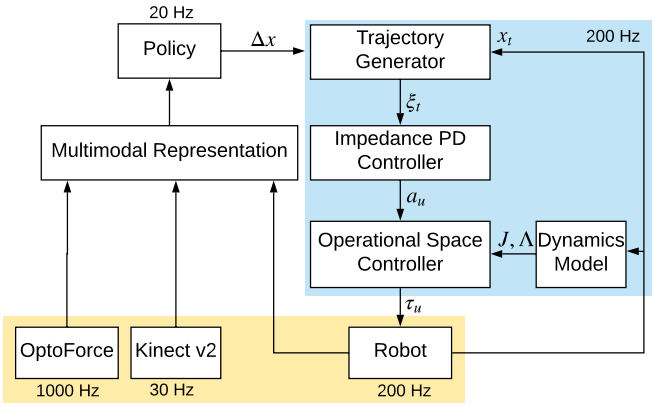


Fig. 3: Our controller takes end-effector position displacements from the policy at 20Hz and outputs robot torque commands at 200Hz. The trajectory generator interpolates high-bandwidth robot trajectories from low-bandwidth policy actions, the impedance PD controller tracks the interpolated trajectory, and the operational space controller uses the robot dynamics model to transform Cartesian-space accelerations into commanded joint torques. The resulting controller is compliant and reactive.

Policy Learning. Modeling contact interactions and multi-contact planning still result in complex optimization problems [35, 36, 45] that remain sensitive to inaccurate actuation and state estimation. We formulate contact-rich manipulation as a model-free reinforcement learning problem to investigate its performance when relying on multimodal feedback and when acting under uncertainty in geometry, clearance and configuration. By choosing model-free, we also eliminate the need of an accurate dynamics model, which is typically difficult to obtain in the presence of rich contacts. Specifically, we choose trust-region policy optimization (TRPO) [39]. TRPO imposes a bound of KL-divergence for each policy update by solving a constrained optimization problem, which prevents the policy from moving too far away from the previous step. The policy network is a 2-layer MLP that takes as input the 128-d multimodal representation and produces a 3D displacement $\Delta\mathbf{x}$ of the robot end-effector. To train the policy efficiently, we freeze the representation model parameters during policy learning, such that it reduces the number of learnable parameters to 3% of the entire model and substantially improves the sample efficiency.

Controller Design. Our controller takes in Cartesian end-effector displacements $\Delta\mathbf{x}$ from the policy at 20Hz, and outputs direct torque commands τ_u to the robot at 200Hz. Its architecture can be split into three parts: trajectory generation, impedance control and operational space control (see Fig 3). Our policy outputs Cartesian control commands instead of joint-space commands, so it does not need to implicitly learn the non-linear and redundant mapping between 7-DoF joint space and 3-DoF Cartesian space. We use direct torque control as it gives our robot compliance during contact, which makes the robot safer to itself, its environment, and any nearby human operator. In addition, compliance makes the peg insertion task easier to accomplish under position uncertainty, as the robot can slide on the surface of the box while pushing downwards [16, 22, 37].

The trajectory generator bridges low-bandwidth output of the policy (which is limited by a forward pass of our representation model), and the high-bandwidth torque control of the robot. Given $\Delta\mathbf{x}$ from the policy and the current end-effector position $\mathbf{x}_t$, we calculate the desired end-effector position $\mathbf{x}_{\text{des}}$. The trajectory generator interpolates between $\mathbf{x}_t$ and $\mathbf{x}_{\text{des}}$ to yield a trajectory $\xi_t = \{\mathbf{x}_k, \mathbf{v}_k, \mathbf{a}_k\}_{k=t}^{t+T}$ of end-effector position, velocity and acceleration at 200Hz. This forms the input to a PD impedance controller to compute a task space acceleration command: $\mathbf{a}_u = \mathbf{a}_{\text{des}} - \mathbf{k}_{\text{p}}(\mathbf{x} - \mathbf{x}_{\text{des}}) - \mathbf{k}_{\text{v}}(\mathbf{v} - \mathbf{v}_{\text{des}})$, where $\mathbf{k}_{\text{p}}$ and $\mathbf{k}_{\text{v}}$ are manually tuned gains.

By leveraging the kinematic and dynamics models of the robot, we can calculate joint torques from Cartesian space accelerations with the dynamically-consistent operational space formulation [24]. The force at the end-effector is calculated with $\mathbf{F} = \Lambda\mathbf{a}_u$, where Λ is the inertial matrix in the end-effector frame that decouples the end-effector motions. Finally, we map from $\mathbf{F}$ to joint torque commands with the Jacobian J: $\tau_{\text{u}} = J^T(\mathbf{q})\mathbf{F}$.

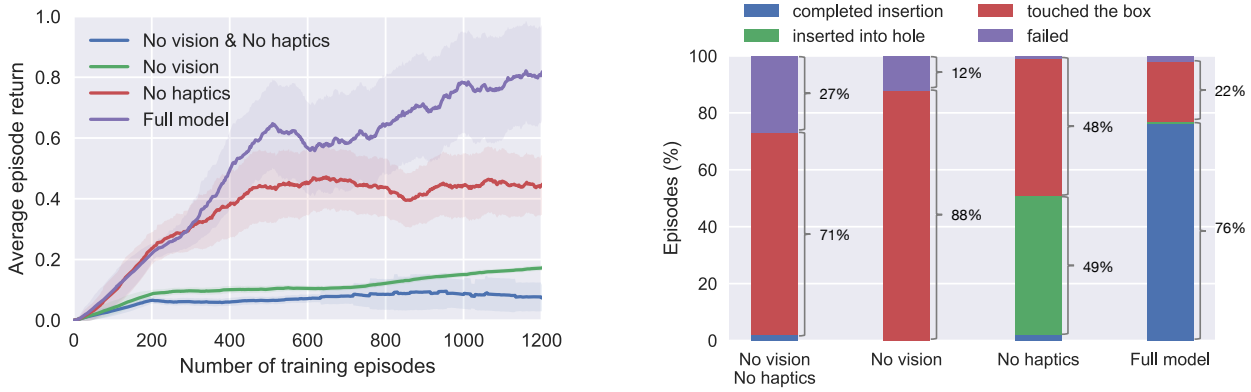
VI. Experiments: Design and Setup

The primary goal of our experiments is to examine the effectiveness of the multimodal representations in contact-rich manipulation tasks. In particular, we design the experiments to answer the following three questions: 1) What is the value of using all modalities *simultaneously* as opposed to 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?

Task Setup. We design a set of peg insertion tasks where task success requires joint reasoning over visual and haptic feedback. We use five different types of pegs and holes fabricated with a 3D printer: round peg, square peg, triangular peg, semicircular peg, and hexagonal peg, each with a nominal clearance of around 2mm as shown in Figure 5a.

Robot Environment Setup. For both simulation and real robot experiments, we use the Kuka LBR IIWA robot, a 7-DoF torque-controlled robot. Three sensor modalities are available in both simulation and real hardware, including proprioception, an RGB camera, and a force-torque sensor. The proprioceptive feature is the end-effector pose as well as linear and angular velocity. They are computed using forward kinematics. RGB images are recorded from a fixed camera pointed at the robot. Input images to our model are down-sampled to 128×128. We use the Kinect v2 camera on the real robot. In simulation, we use CHAI3D [13] to render the graphics. The force sensor provides a 6-axis feedback that measures both the force and the moment along the x,y,z axes. On the real robot, we mount an OptoForce sensor between the last joint and the peg. In simulation, the contact between the peg and the box is modeled with SAI 2.0 [14], a real-time physics simulator for rigid articulated bodies with high fidelity contact resolution.

Reward Design. We use the following staged reward function to guide the reinforcement learning algorithm through the different sub-tasks, simplifying the challenge of explo-



(a) Training curves of reinforcement learning

(b) Policy evaluation statistics

Fig. 4: Simulated Peg Insertion: Ablative study of representations trained on different combinations of sensory modalities. We compare our full model, trained with a combination of visual and haptic feedback and proprioception, with baselines that are trained without vision, or haptics, or neither. (b) The graph shows partial task completion rates with different feedback modalities, and we note that both the visual and haptic modalities play an integral role for contact-rich tasks.

ration and improving learning efficiency:

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

where $\mathbf{s} = (s_x, s_y, s_z)$ and $\mathbf{s}_{xy} = (s_x, s_y)$ use the peg's current position, λ is a constant factor to scale the input to the tanh function, the target peg position is $(0, 0, -h_d)$ where h_d is the height of the hole, and c_r and c_a are constant scale factors.

Evaluation Metrics. We report the quantitative performances of the policies using the sum of rewards achieved in an episode, normalized by the highest attainable reward. We also provide the statistics of the stages of the peg insertion task that each policy can achieve, and report the percentage of evaluation episodes in the following four categories:

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 makes contact with the box but no insertion is achieved;
4) *failed*: the peg fails to reach the box.

Implementation Details. To train each representation model, we collect a multimodal dataset of 100k states and generate the self-supervised annotations. We roll out a random policy as well as a heuristic policy while collecting the data, which encourages the peg to make contact with the box. The representation models are trained for 20 epochs on a Titan V GPU before taking to the policy learning.

VII. Experiments: Results

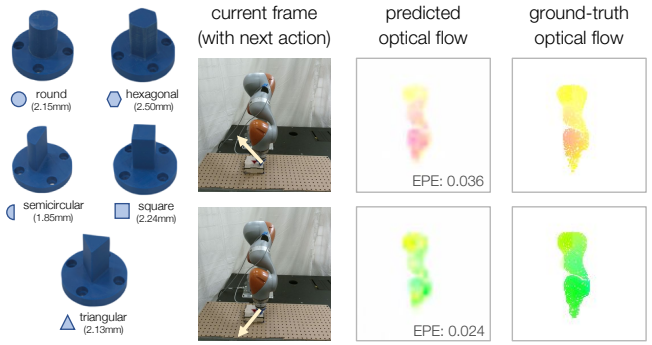
We first conduct an ablative study in simulation to investigate the contributions of individual sensory modalities to learning the multimodal representation and manipulation policy. We then apply our full multimodal model to a real robot, and train reinforcement learning policies for the peg insertion tasks from the learned representations with high sample efficiency. Furthermore, we visualize the representations and provide a detailed analysis of robustness with respect to shape and clearance variations.

A. Simulation Experiments

Peg insertion requires the controller to leverage the synergy between multisensory inputs. The visual feedback guides the arm to reach the box from its initial position. Once contact is made with the box, the haptic feedback guides the end-effector to insert the peg. As shown in Figure 2, three modalities are jointly encoded by our representation model, including RGB images, force readings, and proprioception. Here, we investigate the importance of these sensory modalities for contact-rich manipulation tasks. Therefore, we perform an ablative study in simulation, where we learn the multimodal representations with different combinations of modalities. These learned representations are subsequently fed to the TRPO policies to train on a task of inserting a square peg. We randomize the configuration of the box position and the arm's initial position at the beginning of each episode to enhance the robustness and generalization of the model.

We illustrate the training curves of the TRPO agents in Figure 4a. We train all policies with 1.2k episodes, each lasting 500 steps. We evaluate 10 trials with the stochastic policy every 10 training episodes and report the mean and standard deviation of the episode rewards. Our `Full model` corresponds to the multimodal representation model introduced in Section IV, which takes all three modalities as input. We compare it with three baselines: `No vision` masks out the visual input to the network, `No haptics` masks out the haptic input, and `No vision No haptics` leaves only proprioceptive input. From Figure 4a we observe that the absence of either visual and force modality negatively affects task completion, with `No vision No haptics` performing the worst. None of the three baselines has reached the same level of performance as the final model. Among these three baselines, we see that the `No haptics` baseline achieved the highest rewards. We hypothesize that vision locates the box and the hole, which facilitates the first steps of robot reaching and peg alignment, while haptic feedback is uninformative until after contact is made.

The `Full model` achieves the highest success rate with nearly 80% completion rate, while all baseline methods



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

Fig. 5: (a) 3D printed pegs used in the real robot experiments and their box clearances. (b) Qualitative predictions: We visualize examples of optical flow predictions from our representation model (using color scheme in [18]). The model predicts different flow maps on the same image conditioned on different next actions indicated by projected arrows.

have a completion rate below 5%. It is followed by the `No haptics` baseline, which relies solely on the visual feedback. We see that it is able to localize the hole and perform insertion half of the time from only the visual inputs; however, few episodes have completed the full insertion. It implies that the haptic feedback plays a more crucial role in determining the actions when the peg is placed in the hole. The remaining two baselines can often reach the box through random exploration, but are unable to exhibit consistent insertion behaviors.

B. Real Robot Experiments

We evaluate our `Full model` on the real hardware with round, triangular, and semicircular pegs. In contrast to simulation, the difficulty of sensor synchronization, variable delays from sensing to control, and complex real-world dynamics introduce additional challenges on the real robot. We make the task tractable on a real robot by training a shallow neural network controller while freezing the multimodal representation model that can generate action-conditional flows with low endpoint errors (see Figure 5b).

We train the TRPO policies for 300 episodes, each lasting 1000 steps, roughly 5 hours of wall-clock time. We evaluate each policy for 100 episodes in Figure 6. The first three bars correspond to the set of experiments where we train a specific representation model and policy for each type of peg. The robot achieves a level of success similar to that in simulation. A common strategy that the robot learns is to reach the box, search for the hole by sliding over the surface, align the peg with the hole, and finally perform insertion. More qualitative behaviors can be found in the supplementary video.

We further examine the potential of transferring the learned policies and representations to two novel shapes previously unseen in representation and policy training, the hexagonal peg and the square peg. For policy transfer, we take the representation model and the policy trained for the triangular peg, and execute with the new pegs. From the 4th and 5th bars in Figure 6, we see that the policy achieves over 60% success rate on both pegs without any further policy training on them. A better transfer performance can

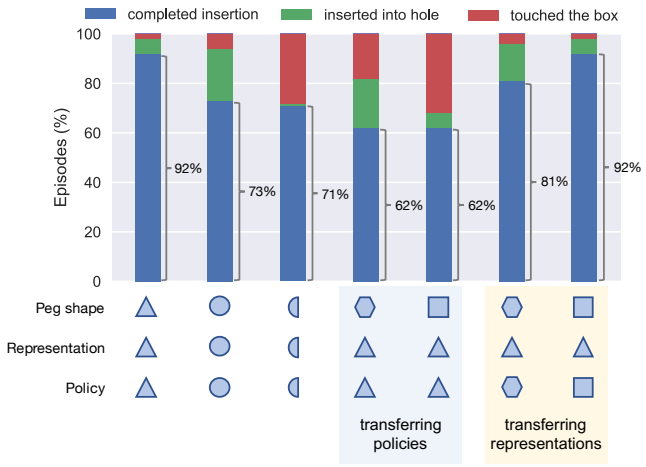


Fig. 6: Real Robot Peg Insertion: We evaluate our `Full Model` on the real hardware with different peg shapes, indicated on the x-axis. The learned policies achieve the tasks with a high success rate. We also study transferring the policies and representations from trained pegs to novel peg shapes (last four bars). The robot effectively re-uses previously trained models to solve new tasks.

be achieved by taking the representation model trained on the triangular peg, and training a new policy for the new pegs. As shown in the last two bars in Figure 6, the resulting performance increases 19% for the hexagonal peg and 30% for the square peg. Our transfer learning results indicate that the multimodal representations from visual and haptic feedback generalize well across variations of our contact-rich manipulation tasks.

Finally, we study the robustness of our policy in the presence of sensory noise and external perturbations to the arm by periodically occluding the camera and pushing the robot arm during trajectory roll-out. The policy is able to recover from both the occlusion and perturbations. Qualitative results can be found in our supplementary video on our website: `https://sites.google.com/view/visionandtouch`.

VIII. Discussion and Conclusion

We examined the value of jointly reasoning over time-aligned multisensory data for contact-rich manipulation tasks. To enable efficient real robot training, we proposed a novel model to encode heterogeneous sensory inputs into a compact multimodal representation. Once trained, the representation remained fixed when being used as input to a shallow neural network policy for reinforcement learning. We trained the representation model with self-supervision, eliminating the need for manual annotation. Our experiments with tight clearance peg insertion tasks indicated that they require the multimodal feedback from both vision and touch. We further demonstrated that the multimodal representations transfer well to new task instances of peg insertion. For future work, we plan to extend our method to other contact-rich tasks, which require a full 6-DoF controller of position and orientation. We would also like to explore the value of incorporating richer modalities, such as depth and sound, into our representation learning pipeline, as well as new sources of self-supervision.

ACKNOWLEDGMENT

This work has been partially supported by JD.com American Technologies Corporation ("JD") under the SAIL-JD AI Research Initiative and by the Toyota Research Institute ("TRI"). This article solely reflects the opinions and conclusions of its authors and not of JD, any entity associated with JD.com, TRI, or any entity associated with Toyota. We are immensely grateful to Oussama Khatib for lending us the Kuka IIWA for the project. We also want to thank Shameek Ganguly and Mikael Jorda for their assistance with the robot controller design and the SAI 2.0 simulator, as well as their many insights during research discussions.

REFERENCES

[1] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics", in *Advances in Neural Information Processing Systems*, 2016, pp. 5074–5082.

[2] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction", *arXiv preprint arXiv:1710.11252*, 2017.

[3] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps", in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1554–1560.

[4] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, "A probabilistic framework for task-oriented grasp stability assessment", in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3040–3047.

[5] A Bicchi, M Bergamasco, P Dario, and A Fiorillo, "Integrated tactile sensing for gripper fingers", in *Int. Conf. on Robot Vision and Sensory Control*, 1988.

[6] R. Blake, K. V. Sobel, and T. W. James, "Neural synergy between kinetic vision and touch", *Psychological science*, vol. 15, no. 6, pp. 397–402, 2004.

[7] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action", *IEEE Transactions on Robotics*, vol. 33, pp. 1273–1291, Dec. 2017.

[8] T. de Bruin, J. Kober, K. Tuyls, and R. Babuška, "Integrating state representation learning into deep reinforcement learning", *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1394–1401, 2018.

[9] 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.

[10] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch", *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[11] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The feeling of success: Does touch sensing help predict grasp outcomes?", *Conference on Robot Learning (CoRL)*, 2017.

[12] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, and S. Levine, "Path integral guided policy search", in *ICRA*, 2017.

[13] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, J. Warren, O. Khatib, and K. Salisbury, "The chai libraries", in *Proceedings of Eurohaptics 2003*, Dublin, Ireland, 2003, pp. 496–500.

[14] F. Conti and O. Khatib, "A framework for real-time multi-contact multi-body dynamic simulation", in *Robotics Research*, Springer, 2016, pp. 271–287.

[15] G. M. Edelman, *Neural Darwinism: The theory of neuronal group selection.* Basic books, 1987.

[16] C. Eppner, R. Deimel, J. Álvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping", *Int. J. Rob. Res.*, vol. 34, no. 7, pp. 1021–1038, Jun. 2015.

[17] C. Finn and S. Levine, "Deep visual foresight for planning robot motion", in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, pp. 2786–2793.

[18] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks", *CoRR*, vol. abs/1504.06852, 2015. arXiv: 1504.06852.

[19] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 536–543.

[20] C. Garcia Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation", *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 2, pp. 577–584, Apr. 2017.

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

[22] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning force control policies for compliant manipulation", in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4639–4644.

[23] D. Kappler, P. Pastor, M. Kalakrishnan, M. Wuthrich, and S. Schaal, "Data-driven online decision making for autonomous manipulation", in *Proceedings of Robotics: Science and Systems*, Rome, Italy, 2015.

[24] O. Khatib, "Inertial Properties in Robotic Manipulation: An Object-Level Framework", *Int. J. Rob. Res.*, vol. 14, no. 1, pp. 19–36, 1995. arXiv: 9809069v1 [arXiv:gr-qc].

[25] S. Lacey and K. Sathian, "Crossmodal and multisensory interactions between vision and touch", in *Scholarpedia of Touch*, Springer, 2016, pp. 301–315.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *nature*, vol. 521, no. 7553, p. 436, 2015.

[27] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat., "State representation learning for control: An overview", *CoRR*, vol. abs/1802.04181, 2018.

[28] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies", *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, Jan. 2016.

[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning", *arXiv preprint arXiv:1509.02971*, 2015.

[30] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor, "Learning end-to-end multimodal sensor policies for autonomous navigation", *arXiv preprint arXiv:1705.10422*, 2017.

[31] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning", in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[32] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games", in *Advances in neural information processing systems*, 2015, pp. 2863–2871.

[33] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio", *arXiv preprint arXiv:1609.03499*, 2016.

[34] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features", *ECCV*, 2018.

[35] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation", 2016, pp. 842–849.

[36] M Posa, C Cantu, and R Tedrake, "A direct method for trajectory optimization of rigid bodies through contact", *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1044–1044, Jun. 2014.

[37] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization", *Autonomous Robots*, vol. 36, no. 1, pp. 11–30, 2014.

[38] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing", *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.

[39] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization", in *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[40] J. Sinapov, C. Schenck, and A. Stoytchev, "Learning relational object categories using behavioral exploration and multimodal percep-

tion", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 5691–5698.

[41] H. Song, Y. Kim, and J. Song, "Automated guidance of peg-in-hole assembly tasks for complex-shaped parts", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4517–4522.

[42] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines", in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[43] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor", in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 297–303.

[44] 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.

[45] S. Tonneau, A. D. Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots", *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[46] F. Veiga, H. Van Hoof, J. Peters, and T. Hermans, "Stabilizing novel objects by learning to predict tactile slip", in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 5065–5072.

[47] D. E. Whitney, "Quasi-Static Assembly of Compliantly Supported Rigid Parts", *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, 1982.

[48] 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.

[49] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo, "Deep multimodal representation learning from temporal data", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5066–5074.

[50] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, *et al.*, "Reinforcement and imitation learning for diverse visuomotor skills", *arXiv preprint arXiv:1802.09564*, 2018.