2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Seq2Seq Imitation Learning for Tactile Feedback-based Manipulation

Wenyan Yang[1], Alexandre Angleraud[1], Roel S. Pieters[1], Joni Pajarinen[2], and Joni-Kristian Kämäräinen[1]

***Abstract*— Robot control for tactile feedback based manipulation can be difficult due to modeling of physical contacts, partial observability of the environment, and noise in perception and control. This work focuses on solving partial observability of contact-rich manipulation tasks as a *Sequence-to-Sequence (Seq2Seq)* Imitation Learning (IL) problem. The proposed Seq2Seq model first produces a robot-environment interaction sequence to estimate the partially observable environment state variables, and then, the observed interaction sequence is transformed to a control sequence for the task itself. The proposed Seq2Seq IL for tactile feedback based manipulation is experimentally validated on a door-open task in a simulated environment and a snap-on insertion task with a real robot. The model is able to learn both tasks from only 50 expert demonstrations while state-of-the-art reinforcement learning and imitation learning methods fail.**

I. INTRODUCTION

The sense of touch is a key sensory modality for many robot manipulation tasks such as grasping [1] and precision-insertion [2], [3], [4]. Tactile sensing is an instrumental modality of robotic manipulation, as it provides information that is not accessible via remote sensors such as cameras or lidars. The key challenges in tactile sensing based control are the difficulty to accurately model physical contacts, partial observability of the environment from touch only, and noise in perception and control. Tactile feedback based manipulation controllers have been proposed, but they often use heuristic search patterns [5] or are tailored for a specific task [3]. In such case, the learning-based approaches are more promising to learn generic solutions for contract-rich manipulation control tasks.

Reinforcement learning (RL) is one of the promising areas of machine learning for robotics. The goal of RL is to learn an optimal policy which maximizes the long-term cumulative rewards. In the case of contact-rich manipulation, RL learning becomes challenging due to sparse reward and partial observability. That often results to an excessive amount of environment interactions needed, which is not doable with real robots. An alternative option is to use simulations for teaching, but the difficulty to accurately model physical contacts becomes the bottleneck. Therefore, instead of pure RL, a more feasible solution is imitation learning (IL). In IL instead of trying to learn from the sparse rewards or manually specifying a reward function, an expert (typically a human) provides a set of demonstrations. The agent then tries to learn the optimal policy by following, imitating the expert's decisions. A number of imitation learning methods have been proposed [6], [7], [8], [9], [10], but these do not particularly address the partial-observability problem that is inherent in tactile sensing.

In this work, we focus on solving contact-rich manipulation tasks with tactile-only sensing and in partially observable environments. Motivated by the above discussion, we aim to design a method that safely and sample-efficiently learns contact-rich tasks with minimal manual engineering. Safety in our case means that IL following expert demonstrations better avoids generating dangerous actions than RL. To address the partial observability, we take the common approach of RL for Partially-observable Markov Decision Processes (POMDPs): history data is used to aggregate belief of the partially observable environment states. We combine it with IL framework to achieve sample efficiency. Two types of expert demonstration are used in this work: exploration and manipulation. The robot first imitates the expert's exploration for hidden state discovery, and use the exploration observations to produce a goal-directed trajectory that imitates the expert's manipulation. The main contributions are:

- A novel Sequence-to-Sequence (Seq2Seq) model to perform exploration-to-manipulation imitation learning. The model learns to translate the exploration trajectory into a manipulation trajectory. Generation of both types of trajectories is learned from expert demonstrations.
- We show that a Transformer based Seq2Seq IL architecture is able to aggregate belief of hidden environment states during exploration. Besides, enforcing the Seq2Seq encoded feature to be similar to the hidden environment states contributes to task performance.
- The Proposed Seq2Seq IL is sample efficient. Sample efficiency is investigated and compared to strong baselines in the both simulated and real manipulation tasks. Seq2Seq IL learns successful control policies from only 50 expert demonstrations.

II. RELATED WORK

The existing controllers for tactile feedback manipulation often use control heuristics and need a model for the contact dynamics [3], [5], [11]. Recently, a number of methods combining force kinematics with compliant control and machine learning have been proposed to overcome the need of manual tuning [12], [13], [14], [15], [16]. Moreover, Imitation Learning (IL), also known as Learning from Demonstration (LfD), allows more flexible "programming" of a robot as it learns from expert demonstrations. Various computational models, including Gaussian Mixture Models, Hidden Markov Models, Deep Neural Networks and Dynamic Motion Primitives, have been proposed for IL [6], [7], [8], [9], [10]. These

[1]Tampere University, Finland; [2]Aalto University, Finland.

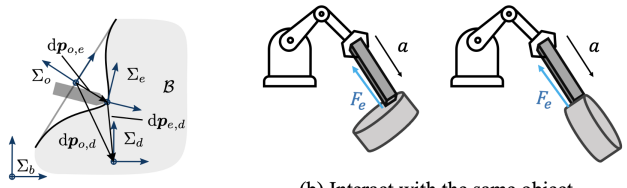
approaches differ from our work rather strongly in the sense that they assume a fully observable environment or a model for the contact physics or are task specific ("Peg-in-Hole" or "goal reaching"). We seek for a generic method that does not need to model the contact physics.

A number of generic RL and RL-based IL methods exist [17], [18], [19], [20], [21], [22], [23], [24]. However, they as well assume a fully observable environment that can be modeled as a Markov Decision Process (MDP), and require a large number of interaction steps to converge. The partial-observability problem (POMDP) has been addressed in a number of works [25], [26], [27], [28], [29], [30], [31], [32], [33]. We adopt the common approach in these works, and use the observation history to aggregate belief of the partially observable state variables. The aggregation can be done by sequential learning using recurrent neural networks [34], [35], [36], [30], [37]. The main shortcoming of the above RL approaches for POMDPs is that they need a massive amount of environment interactions which makes them unsuitable for real robot tasks.

From the imitation learning perspective, a numerous work have proposed to learn a model-free policy for expert demonstrations tasks [38], [39], [40], [41], [42]. However, these methods still requiures online data sampling and is not practically sample-efficient for real applications. Some offline imitation learning methods such as implicit behaviour cloning (IBC) [43], DemoDice [44], etc., but they are not designed for solving POMDP problems.

III. Background

A. Tactile feedback in manipulation



(a) Interaction dynamics

(b) Interact with the same object at different pose

Fig. 1: Illustration of the physical wrench (torques and forces) during robot manipulation: (a) the interaction frames involved in Eq. 1 and (b) an example where the observed wrench is the same for two distinct surface (task) points.

In this work, the contact sensory feedback refers to the external wrench $\mathbf{F}_{ext}$ observed at the robot end-effector. Tactile feedback provides only partial observation of the environments, the object pose in our example. This can be analytically studied via the contact kinematics investigated in mechanical engineering and robotics [45], [46], [47].

Consider a robot-environment interaction model, where the robot interacts with a target surface directly with its end-effector or through an object in its gripper. Suppose the environment is defined by its stiffness, and the end-effector is treated as a rigid object. For rigid objects the roles can be interchanged. Denote Σ_b as the Cartesian base frame, Σ_d is the target object frame, Σ_e is the end-effector frame, and Σ_o is the task frame located on the target object surface. In this setting, the surface provides resistance to the robot end-effector's attempts to penetrate the target. This resistance can be modeled as the external wrench [48]

$$ {}^{o}\mathbf{F}_{ext} = ({}^{o}\mathbf{K}_e + \mathbf{K}_{P,cart})^{-1}\,{}^{o}\mathbf{K}_e\mathbf{K}_{P,cart}\mathrm{d}\mathbf{p}_{o,d} \ , \qquad (1) $$

where the overall stiffness matrix consists of a Cartesian controller stiffness matrix $\mathbf{K}_{P,cart}$ and the environment's resistance-to-penetration stiffness matrix ${}^{o}\mathbf{K}_e$. ${}^{o}\mathbf{K}_e$ is composed by the translational and rotational components, but here we omit the rotation component and assume purely translational stiffness matrices for simplicity.

The translational interaction is given by $\mathrm{d}\mathbf{p}_{o,d}$ that is the displacement of the task frame Σ_o with respect to the target frame Σ_d (Figure 1(a)). The two terms for the analysis are ${}^{o}\mathbf{K}_e$ that depends on the target object's material and $d\mathbf{p}_{o,d}$ that depends on the normal of the object's surface at the task frame Σ_o. These terms verify that the observable wrench in Eq. 1 depends on the surface stiffness matrix (end-effector assumed rigid) and on the surface normals with respect to the end-effector frame Σ_e as illustrated in Fig. 1(a). If we assume homogeneous stiffness properties for the target object that means that the observed wrench is the same for all task frames, surface points Σ_o for which the end-effector-surface normals are equal. This is illustrated for a solid object in Fig. 1(b). *To summarize, it is not possible to infer the object pose from a single touch sensor measurement.*

B. Partially Observable Markov Decision Process

If the underlying environment state cannot be fully ascertain, the problem can be formulated as POMDP [49]. Formally, a POMDP can be described as a 6-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O})$ where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, $\mathcal{R}$, are the states, actions, transitions and rewards. At time t, the environment is in some state $s_t \in S$, and agent generates action $a_t \in A$. The environment produces a new state $s_{t+1} \in S$ based on dynamics $T(s_{t+1}|s_t, a_t)$ and the agent receives a reward $r_t = R(s_t, a_t, s_{t+1})$. However, the agent cannot directly observe the underlying state S in a POMDP. Instead, the agent receives an observation $o_t \in O$ via the indirect observation function $O(s_{t+1}, a_t, o_t) = P(o_t|s_{t+1}, a_t)$. In general, this implies that the agent must take the entire history of observations and actions $h_t = ((a_0, o_0), (a_1, o_1), \ldots, (a_{t-1}, o_{t-1}))$ into account to make the current state more observable [50].

Based on our analysis of the tactile feedback, the uncertainty of the target object pose from touch sensing makes the task environment only *partially observable* for a force-feedback controller. However, the history of tactile feedback helps to estimate the pose which eventually helps to solve the manipulation task.

C. Tactile-only manipulation tasks

In this work, two high-precision manipulation tasks are studied as representations of tactile-only manipulation tasks. The two test environments, as shown in Figure 3, consist of a simulated door-opening task and a real snap-on insertion task:

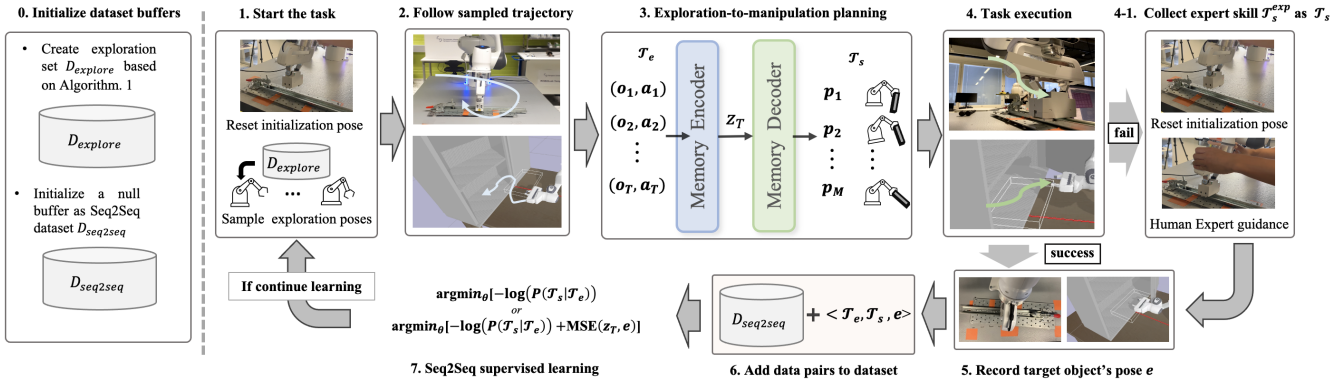
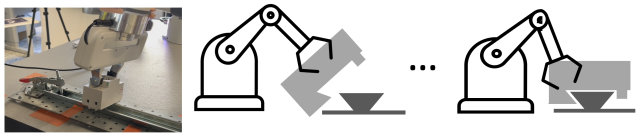
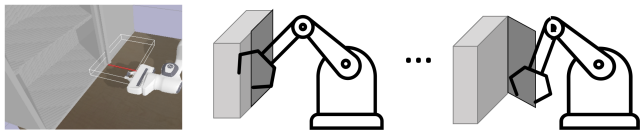


Fig. 2: The proposed Seq2Seq Imitation Learning pipeline. **Step 0**: initialize the datasets. **Step 1-2**: the robot first samples and executes one of the given expert exploration trajectories to collect an observation trajectory $\mathcal{T}_e$. **Step 3-4**: the encoder infers an underlying environment state z from $\mathcal{T}_e$, and then the decoder plans and execute a skill trajectory $\mathcal{T}_s$. If the task fails, **Step 4-1**: the robot returns to initial pose and expert guides the robot to complete the task, meanwhile record expert skill trajectory $\mathcal{T}_s^{exp}$. **Step 6-7**:. the sequence pair $\{\mathcal{T}_e, \mathcal{T}_s, e\}$ are incrementally added to dataset and are used to fine-tune the imitation model. The pipeline stops if the imitation model successfully accomplish the task with 10 continuous tests.



(a) socket snap-on task (real)



(b) Blind door open task (simulation)

Fig. 3: The real (a) and simulated (b) test environments. The relative pose of the installed socket and the assembly rail (a) and the robot-end effector and the cabin door (b) are randomized in the experiments.

a) Simulated door-open task: This task is a customized version of the "Opening" task in the MiniTouch benchmark in [51] and implemented in PyBullet. In our task the cabin is randomly placed in the front of a virtual Franka Emika Panda. The task is to open the cabin door using external wrench feedback and robot's states (pose and velocity of the Emika end-effector).

b) Snap-on mounting task: The real snap-on mounting is a high-precision assembly task where Panda Emika needs to mount an electronic socket into an assempy rail (see Fig. 3). Similar to the above simulated task the robot uses only end-effector's external wrench feedback and end-effector's pose and velocity.

To complete these tasks, the robot must accurately estimate the relative pose of the end-effector with respect to the manipulated object (such as the door edge or assembly rail). The tasks are particularly challenging for tactile-only sensing for two reasons: first, these tasks require high precision, with the snap-on task requiring rotations smaller than one degree and distances smaller than five millimeters, and the door-opening task requiring a relative distance smaller than one centimeter. Second, the relative poses cannot be directly measured by a tactile sensor (as outlined in Section **??**). To successfully complete the manipulation tasks, the robot must explore the environment in order to observe and better understand its state.

IV. Seq2Seq Imitation Learning

Imagine manipulating objects in a dark room. For example, find and pick up your coffee cup. After finding the cup, but before grasping it, you do "tactile exploration" to find its handle and to adjust your grip. Inspired by this strategy, we introduce a model and learning procedures for Sequence-to-Sequence imitation learning (Seq2Seq IL). The robot first executes an exploration trajectory to collect and infer the hidden information of the environment, then it plans a trajectory according to the aggregation of the collected information.

Sequence-to-sequence is a common approach to solve sequential problems [52]. The problem can be defined as sequential mapping of one T-length input sequence $X = \{x_1, x_2, ..., x_T\}$ to an M-length output sequence $Y = \{y_1, y_2, ..., y_M\}$. The overall structure of our Seq2Seq model is illustrated in Fig. 4. The model contains two memory modules which are linked by an encoder-decoder structure. The encoder sequentially receives inputs which are encoded to an internal representation z_t. In our terminology, the encoder performs the environment exploration and produces z_t as a state estimate of partially observable environment. The skill planning is performed by the decoder that auto-regressively generates the target sequence after receiving the encoder's state estimate z_t. The Seq2Seq model can be implemented as an LSTM network [52], or more recently, as a Transformer network [53].

As a distinct feature to other similar works, our Seq2Seq model breaks the tactile feedback manipulation tasks into two stages: 1) *exploration stage* and 2) *skill planning stage*. In the exploration stage the robot follows the encoder trajectory to explore the environment and encode its state into the

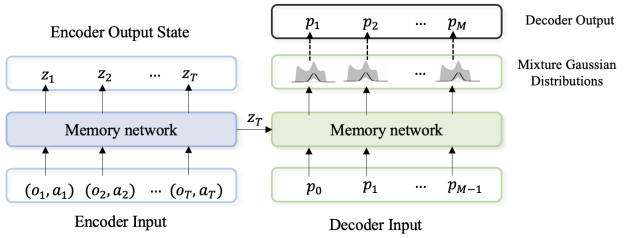


Fig. 4: The overall Seq2Seq architecture used in our model. The input sequence is an exploration trajectory $\mathcal{T}_e = \{(o_t, a_t)\}$ and the output sequence is a skill execution trajectory $\mathcal{T}_s = \{p_m\}$. The encoder performs feature aggregation to infer the current state z_T of a partially observable environment.

internal representation z_t. In the skill planning stage, the decoder produces a goal-directed control trajectory for the low level robot controller to complete the main manipulation task. The both encoder and decoder trajectories are learned via imitation learning, i.e. from expert demonstrations.

A. Sequence modeling

For the exploration stage, a recorded expert exploration control trajectory (via-points) are given to the robot. The robot follows the expert trajectory to collect tactile feedback from the environment. The usage of expert trajectories and the robot in impedance control mode allows safe exploration. $\mathcal{T}_e$ is the observed T-length exploration trajectory,

$$\mathcal{T}_e = \{(o_1, a_1), (o_2, a_2), ..., (o_N, a_T)\} \ , \tag{2}$$

where o_i is an observation containing 18 attributes: end-effector external wrench ${}^o\mathbf{F}_{ext}$ (translational and rotational components), end-effector pose (3D translation and orientation), and its velocity (translational and rotational). a is a 6-dimensional action vector of the end-effector's displacement. The skill planning trajectory (via-points) $\mathcal{T}_s$ is

$$\mathcal{T}_s = \{p_1, p_2, \ldots, p_M\} \ , \tag{3}$$

where p_i is a 6-dimensional end-effector pose.

Given the above definitions of the exploration and skill planning trajectories, the problem can be cast as a Seq2Seq problem. A Seq2Seq controller provides the robot low level controller a task-directed trajectory (a via-point sequence) $\mathcal{T}_s$ from the exploration (and observation) sequence $\mathcal{T}_e$.

B. Seq2Seq encoder-decoder

1) Exploration encoder: In our encoder-decoder structure, the Seq2Seq encoder gradually observes $\mathcal{T}_e$ and encodes the observations into an internal representation z_T. In our formalism we assume that the encoder representation includes the state variables that are only partially observable, but for which belief is aggregated through history data. The encoder is depicted in the left-hand-side of Fig. 4 and is formally

$$z_T = \text{Enc}(\mathcal{T}_e) \tag{4}$$

where z_T is the encoded representation and approximates belief state of the POMDP.

2) Skill planning decoder: The Seq2Seq decoder receives z_T from the encoder and produces the skill plan trajectory $\mathcal{T}_s$. The history of the previous planned poses is $\tau_{t-1} = \{p_0, ..., p_{t-1}\}$. The probability of the next skill planning pose is $P(p_t|z_T, \tau_{t-1})$, and is modeled as a mixture of K Gaussians. The Gaussian mixture probability of planned pose p_t at the step t is defined as

$$P(p_t|z_T, \tau_{t-1}) = \sum_{i=1}^{K} w_\theta^i \mathcal{N}(\mu_\theta^i, \sigma_{i,\theta}^2) \tag{5}$$

where w_θ^i is the weight of the i-th Gaussian, μ_θ^i and $\sigma_{i,\theta}^2$ are the mean and variance of the i-th Gaussian. The parameter θ denotes that these are estimated by a Mixture Density Network (MDN) layer applied at the decoder's head (Fig. 4). The Seq2Seq learning objective is to maximize the probability of a target sequence, or respectively, minimize the negative log-likelihood. The loss can be formulated as:

$$\mathcal{L}_{seq2seq} = -\log P(\mathcal{T}_s|\mathcal{T}_e) = -\sum_{t=1}^{M} \log P(p_t|z_T, \tau_{t-1}) \ . \tag{6}$$

3) Supervised Seq2Seq: If the partially observable state variables are known; let's denote them by e; then these can be embedded to the encoder output z_T and learned supervised manner. A suitable loss is, for example, the standard mean-squared error (MSE) loss. The supervised Seq2Seq loss is

$$\mathcal{L}_{supervised} = \mathcal{L}_{seq2seq} + ||z_T - e||^2 \ . \tag{7}$$

The supervised Seq2Seq IL and Seq2Seq IL are two variants used in our experiments. We denote the supervised version as "Seq2Seq-oracle" as, in general, the partially observable variables are not known.

V. Experiments

A. Settings

1) Demonstrations collection and imitation pipeline: Since our model consists of two kind of trajectories, exploration and skill planning, a small number of initial demonstrations of the exploration are first provided, and then the actual expert skill execution demonstrations are demonstrated in the human-in-the-loop manner. Figure 2 illustrates the complete training procedure.

For exploration, a human expert provides a small number (five in our case) exploration trajectories. The experts were instructed to move the robot hand in the kinesthetic teaching mode such that they "feel" that the socket is in the correct position with respect to the rail. See Algorithm 1 for details.

After the initial exploration demonstrations the target (rail/door) pose is randomized. The robot produces and executes an exploration trajectory and given the exploration feedback ($\mathcal{T}_e$) produces and executes a skill planning trajectory ($\mathcal{T}_s$). If the task is successful, the both trajectories ($\mathcal{T}_e$ and $\mathcal{T}_s$) are added to the training data. If the task fails, the robot position is reversed back to the position just before the skill execution, and an expert provides a successful

skill execution $\mathcal{T}_s^{exp}$. Then ($\mathcal{T}_e$, $\mathcal{T}_s^{exp}$) will be added to training data. The most important expert demonstrations for our method are these human-in-the-loop failure correction demonstrations. We apply DAgger-like style learning [54] to incrementally collect data to learn from the experts (see Fig. 2). Our supplementary material contains video clips about training and testing.

For the real snap-on task all demonstrations are provided manually by a human expert. For the simulated door-open task the exploration trajectories are provided manually by an expert (mouse is used to move the end-effector to the door and then move it on the door surface), and the skill execution demonstrations are provided by a SAC trained controller that fully observes the environment (the cabin 3D pose).

Algorithm 1 Expert exploration trajectories collection

Initialize expert exploration set D_{exp}. skill set D_s. Denote the robot end-effector pose as o.
for $i = 1$ to n **do**
  Initialize robot to its starting pose
  Set robot to the kinesthetic teaching mode
  Randomize target object pose
  Let expert guide end-effector to explore
  Record and store expert trajectory $\tau_{explore} = \{\mathbf{o}_0, \mathbf{o}_1, \ldots, \mathbf{o}_T\}$, where o_t is the end-effector pose
  Add $\tau_{explore}$ to D_{exp}
end for

2) Baseline methods and metrics: We use the following criteria to select the baseline methods: 1) the baseline shall be learning-based methods and 2) the baseline shall be designed to solve the POMDP problems. Based on these criteria, we chose the following baseline methods from multiple method categories. **POMDP Reinforcement Learning (POMDP-RL):**

- Recurrent model-free RL using SAC or TD3 [55]: RMF-RL(sac) and RMF-RL(td3)
- RMF-RL modified by adding the Behavior Cloning (BC) [56]: RMF-RL-IL(sac) and RMF-RL-IL(td3)
- Soft-Actor-Critic [57] that observes the full environment (with "oracle"): SAC-MDP

Imitation Learning (IL):

- Soft-Q Imitation Learning is a RL-based IL which assigns sparse rewards to expert demonstrations [58] (due to the POMDP setting, the RL part is RMF-RL(sac)): SQIL
- A classical behavior cloning (BC) [59] where an LSTM network is used to modernize it: BC-LSTM

Our Seq2Seq IL variants:

- Seq2Seq - a Transformer Seq2Seq IL that learns the partially observable state variables through exploration ($\mathcal{L}_{seq2seq}$ in Eq. 6)
- Seq2Seq-oracle - a Transformer Seq2Seq IL that is trained supervised manner ($\mathcal{L}_{supervised}$ in Eq. 7)
- Seq2Seq-LSTM - Otherwise similar to Seq2Seq IL, but the Transfomer is replaced by LSTM from [34]

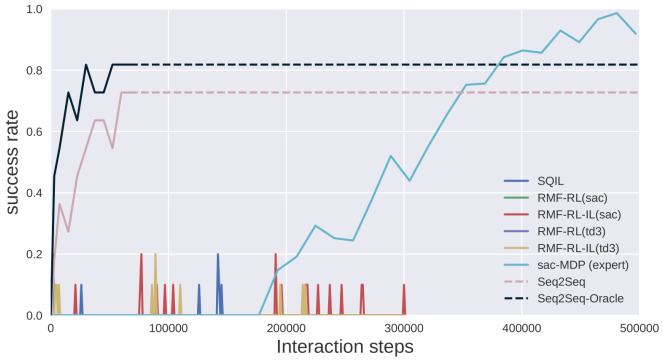


Fig. 5: Sample efficiency experiment (simulated door-open task): success rates as the function of interaction steps. Our methods were trained only for 50 demonstrations (approx. 67k interactions).

In all experiments, the length of the encoder output z_T is set to 3 as it corresponds to the number of actual partially observable task variables, the xy-place location of the target and its rotation.

3) Success rate: Success rate is used as the performance metric. Success rate reports the proportion of test runs where the agent reached the goal state (door opened / socket mounted).

B. Results

1) Sample efficiency: Since sample efficiency is one of the main limiting factor in using machine learning techniques in real robot learning tasks, the first experiment evaluates sample efficiency of the proposed model and compares it to the POMDP RL and IL baselines. The experiment was conducted using the simulated door-open task to allow a large number of samples and to avoid failures that would damage the real robot. 50 expert demonstrations were given to RMF-RL-IL and SQIL. Seq2Seq and Seq2Seq-Oracle were trained using DAgger style and stopped after 50 demonstrations. Figure 5 presents the results.

We used an oracle trained soft-actor-critic (SAC-MDP) as the expert. The oracle SAC-MDP agent observes the door pose, robot's end-effector external wrench, pose, and velocity. SAC-MDP successfully learns the task, but requires more than 400k interactions. It is worth noting that none of the POMDP RL baselines (RMF-RL(sac/td3)) was able to solve the problem even after 500k interactions. The imitation RL methods (SQIL and RMF-RL-IL(sac/td3)) achieved 20% success rate, while their performance was unstable. Using DAgger-style incremental learning, our proposed model (Seq2Seq) achieved 76% success rate.

Our Seq2Seq models were the only to learn the task without oracle (POMDP setting) and required only 50 demonstrations that corresponds to approximately 67k interaction steps. This is clearly better than the oracle SAC-MDP that had fully observable environment. The supervised variant of our method, Seq2Seq-Oracle, obtained 16% higher success rate (82%) than the POMDP Seq2Seq. This indicates that the model can effectively discover the partially observable state variables even without supervision.

2) Performance evaluation: We further evaluated IL methods' performance (success rates). 50 demonstrations were used to train each of them. The results are in Table I (100 random tests for each).

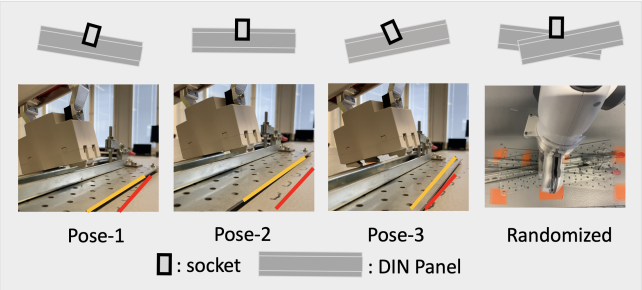


Fig. 6: Examples of the poses used in the robustness and repeatability experiment (the real setup)

SQIL failed on the snap-on task with the real robot and achieved only 20% success rate in the simulated door-open task. BC-LSTM also had poor performance on the both tasks. Seq2Seq-LSTM performed better than the plain BC-LSTM on the simulated door-open task, but completely failed on the real snap-on task. The Seq2Seq model has the second-best performance, 76% success rate on the simulated and 84% on the real snap-on task. This verifies that 1) the Seq2Seq imitation model can learn POMDP tasks effectively, and 2) the Transfomer-based Seq2Seq imitation model is more robust across the tasks than LSTM-based. Our proposed Seq2Seq-Oracle obtained the best performance in both tasks.

3) Detailed analysis on the real robot: We selected the Seq2Seq-Oracle to conduct a more detailed experiments on the snap-on task with a real robot.

a) Robustness to partially observable state variables: To evaluate the robustness of the proposed model we conducted repeatability and robustness experiments. For the repeatability test we executed the learned model 20 times for the rail in a fixed position (Pose-1, Pose-2 and Pose-3, see Fig. 6). To evaluate robustness of the estimation of the partially observable state variables (pose of the rail), we executed the model 102 times for random poses.

The results for the repeatability and robustness experiments are in Table II. The Seq2Seq IL model succeeded 84 out of the 102 attempts with the random rail poses. This indicates fairly good robustness of the method to estimate the non-directly observable state variables. For the two fixed poses, Pose-1 and Pose-2, the repeatability was 100% and 90% (18/20) for Pose-3. These results indicate that the model learns to complete the task, is robust to environment changes and insensitive to observation noise.

b) Number of expert demonstrations: In order to test how effectively the model learns from new expert demonstrations, we recorded 200 expert demonstrations and randomly formed training sets of 5 to 200 demonstrations. These training sets were used to train the model and then the model was tested 45 times. The results are shown in Fig. 7(a). The results verify that the performance rapidly improves and converges after 50 expert demonstrations.

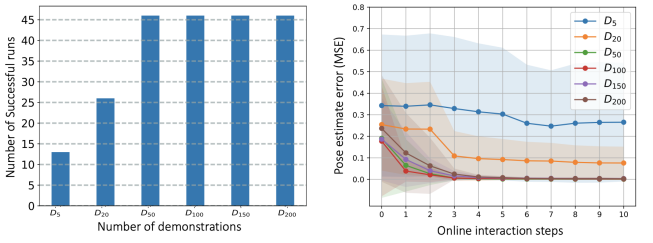
TABLE I: Imitation learning method comparison.

Env.	Exp.*	SQIL	BC-LSTM	Seq2Seq-LSTM	Seq2Seq	Seq2Seq-Oracle
Door-open	95%	20%	11%	56%	76%	89%
Snap-on	100%	-	5%	0%	84%	88%

* 500k trained SAC for door-open and a human expert for snap-on

TABLE II: Repeatability/robustness results for the real robot task (snap-on)

	Randomized	Pose-1	Pose-2	Pose-3
Success	84/102	20/20	20/20	18/20



(a) Successful rate (b) Online environment state estimation

Fig. 7: (a) number of successful test attempts for the model trained with different number of expert demonstrations and (b) Online error in the partially observable state variable estimation (pose) for the same models.

c) Online state estimation: In the final experiment we investigated the exploration part of the Seq2Seq model. Using Seq2Seq-oracle we can obtain current estimates of the partially observable state variables during exploration. In the snap-on task, the variables are the pose of the rail defined by xy-plane coordinates and the rotation angle. We normalized translations and angles to the approximately same scale and computed the error (MSE) online after each exploration step. The results are in Fig. 7(b) and verify that only 4 exploration steps are needed to obtain an accurate estimate of the state variables. This result holds if at least 50 expert demonstrations ($D_{\geq 50}$) are used to train the model.

VI. Conclusions

This work investigated the partial observability problem in learning control policy for tactile-feedback based manipulation. We proposed a transformer-based *Seq2Seq* imitation learning (IL) which imitates expert exploration trajectories, and from them plans a suitable skill trajectory to complete the task. The two stages of Seq2Seq, exploration and skill planning, are learned from expert demonstrations. The proposed model is sample efficient and learns to solve a real snap-on task from only 50 expert demonstrations while the other POMDP RL and IL methods failed. For our future work, we will adapt Seq2Seq IL for closed-loop control, which can lead to better online adaptation. Although we needed to introduce human-in-the-loop learning from expert demonstrations, it also produced substantial boost in sample efficiency, and that opens an intriguing research direction of multi-stage imitation learning.

REFERENCES

[1] Y. Bekiroglu, J. Laaksonen, J. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3.

[2] H. Nakagaki, K. Kitagaki, and H. Tsukune, "Study of insertion task of a flexible beam into a hole," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1999.

[3] Y. Yamamoto, T. Yoneyama, T. Hashimoto, T. Okubo, and T. Itoh, "Sensor-based analysis of high-precision insertion tasks," in *IROS*, 2002.

[4] Y. Ma, D. Xu, and F. Qin, "Efficient insertion control for precision assembly based on demonstration learning and reinforcement learning," *IEEE Trans. on Industrial Informatics*, 2021.

[5] F. Suárez-Ruiz and Q.-C. Pham, "A framework for fine robotic assembly," in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 421–426, IEEE, 2016.

[6] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[7] S. Calinon and A. G. Billard, "What is the teacher's role in robot programming by demonstration?: Toward benchmarks for improved learning," *Interaction Studies*, vol. 8, no. 3, pp. 441–464, 2007.

[8] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 255–262, 2007.

[9] A. G. Billard, S. Calinon, and R. Dillmann, "Learning from humans," *Springer handbook of robotics*, pp. 1995–2014, 2016.

[10] M. Racca, J. Pajarinen, A. Montebelli, and V. Kyrki, "Learning in-contact control strategies from demonstration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 688–695, IEEE, 2016.

[11] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5844–5850, IEEE, 2019.

[12] A. Ranjbar, N. A. Vien, H. Ziesche, J. Boedecker, and G. Neumann, "Residual feedback learning for contact-rich manipulation tasks with uncertainty," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2383–2390, IEEE, 2021.

[13] N. Vuong, H. Pham, and Q.-C. Pham, "Learning sequences of manipulation primitives for robotic assembly," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4086–4092, IEEE, 2021.

[14] Y. Dong, T. Ren, D. Wu, and K. Chen, "Compliance control for robot manipulation in contact with a varied environment based on a new joint torque controller," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 1, pp. 79–90, 2020.

[15] K. Kutsuzawa, S. Sakaino, and T. Tsuji, "Sequence-to-sequence model for trajectory planning of nonprehensile manipulation including contact model," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3606–3613, 2018.

[16] W. Si, Y. Guan, and N. Wang, "Adaptive compliant skill learning for contact-rich manipulation with human in the loop," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5834–5841, 2022.

[17] J. Xu, Z. Hou, W. Wang, B. Xu, K. Zhang, and K. Chen, "Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1658–1667, 2018.

[18] T. Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, and S. Levine, "Offline meta-reinforcement learning for industrial insertion," *arXiv preprint arXiv:2110.04276*, 2021.

[19] O. Spector and M. Zacksenhouse, "Learning contact-rich assembly skills using residual admittance policy," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6023–6030, IEEE.

[20] S. A. Khader, H. Yin, P. Falco, and D. Kragic, "Stability-guaranteed reinforcement learning for contact-rich manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 1–8, 2020.

[21] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," *Advances in neural information processing systems*, vol. 27, 2014.

[22] A. A. Apolinarska, M. Pacher, H. Li, N. Cote, R. Pastrana, F. Gramazio, and M. Kohler, "Robotic assembly of timber joints using reinforcement learning," *Automation in Construction*, vol. 125, p. 103569, 2021.

[23] O. Spector and M. Zacksenhouse, "Deep reinforcement learning for contact-rich skills using compliant movement primitives," *arXiv preprint arXiv:2008.13223*, 2020.

[24] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2062–2069, IEEE, 2018.

[25] J. A. Bagnell, A. Y. Ng, and J. G. Schneider, "Solving uncertain markov decision processes," 2001.

[26] M. Kwon, S. Daptardar, P. R. Schrater, and X. Pitkow, "Inverse rational control with partially observable continuous nonlinear dynamics," *Advances in neural information processing systems*, vol. 33, pp. 7898–7909, 2020.

[27] T. Gangwani, J. Lehman, Q. Liu, and J. Peng, "Learning belief representations for imitation learning in pomdps," in *Uncertainty in Artificial Intelligence*, pp. 1061–1071, PMLR, 2020.

[28] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter, "Rudder: Return decomposition for delayed rewards," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] Z. Ren, R. Guo, Y. Zhou, and J. Peng, "Learning long-term reward redistribution via randomized return decomposition," *arXiv preprint arXiv:2111.13485*, 2021.

[30] L. Meng, R. Gorbet, and D. Kulić, "Memory-based deep reinforcement learning for pomdps," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5619–5626, IEEE, 2021.

[31] Z. Yang and H. Nguyen, "Recurrent off-policy baselines for memory-based continuous control," *arXiv preprint arXiv:2110.12628*, 2021.

[32] D. Han, K. Doya, and J. Tani, "Variational recurrent models for solving partially observable control tasks," *arXiv preprint arXiv:1912.10703*, 2019.

[33] G. Singh, S. Peri, J. Kim, H. Kim, and S. Ahn, "Structured world belief for reinforcement learning in pomdp," in *International Conference on Machine Learning*, pp. 9744–9755, PMLR, 2021.

[34] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *AAAI Fall Symposium*, 2015.

[35] M. Igl, L. Zintgraf, T. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for POMDPs," in *Int. Conf. on Machine Learning (ICML)*, 2018.

[36] A. Lee, A. Nagabandi, P. Abbeel, and S. Levine, "Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model," in *Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

[37] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for pomdps," in *International Conference on Machine Learning*, pp. 2117–2126, PMLR, 2018.

[38] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[39] S. Desai, I. Durugkar, H. Karnan, G. Warnell, J. Hanna, and P. Stone, "An imitation from observation approach to transfer learning with dynamics mismatch," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3917–3929, 2020.

[40] G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim, "Demodice: Offline imitation learning with supplementary imperfect demonstrations," in *International Conference on Learning Representations*, 2022.

[41] R. Dadashi, L. Hussenot, M. Geist, and O. Pietquin, "Primal wasserstein imitation learning," *arXiv preprint arXiv:2006.04678*, 2020.

[42] W. Yang, N. Strokina, J. Pajarinen, and J.-k. Kamarainen, "Constrained imitation q-learning with earth mover's distance reward," in *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

[43] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*, pp. 158–168, PMLR, 2022.

[44] G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim, "Demodice: Offline imitation learning with supplementary imperfect demonstrations," in *International Conference on Learning Representations*, 2022.

[45] N. P. Garg, D. Hsu, and W. S. Lee, "Learning to grasp under uncertainty using POMDPs," in *International Conference on Robotics and Automation (ICRA)*, pp. 2751–2757, IEEE, 2019.

[46] D. Montana, "The kinematics of contact and grasp," *The International Journal of Robotics Research*, vol. 7, no. 3, 1988.
[47] S. Howard, M. Zefran, and V. Kumar, "On the 6×6 cartesian stiffness matrix for three-dimensional motions," *Mechanism and Machine Theory*, vol. 33, no. 4, 1998.
[48] T. Gold, A. Volz, and K. Graichen, "Model Predictive Position and Force Trajectory Tracking Control for Robot-Environment Interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
[49] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
[50] H. Nguyen, B. Daley, X. Song, C. Amato, and R. Platt, "Belief-grounded networks for accelerated robot learning under partial observability," *arXiv preprint arXiv:2010.09170*, 2020.
[51] S. Rajeswar, C. Ibrahim, N. Surya, F. Golemo, D. Vazquez, A. Courville, and P. O. Pinheiro, "Haptics-based curiosity for sparse-reward tasks," in *Conference on Robot Learning*, pp. 395–405, PMLR, 2022.
[52] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.
[53] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
[54] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, JMLR Workshop and Conference Proceedings, 2011.
[55] T. Ni, B. Eysenbach, and R. Salakhutdinov, "Recurrent model-free rl can be a strong baseline for many pomdps," in *Int. Conf. on Machine Learning (ICML)*, 2022.
[56] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299, IEEE, 2018.
[57] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.
[58] S. Reddy, A. D. Dragan, and S. Levine, "Sqil: Imitation learning via reinforcement learning with sparse rewards," *arXiv preprint arXiv:1905.11108*, 2019.
[59] M. Bain and C. Sammut, "A framework for behavioural cloning.," in *Machine Intelligence 15*, pp. 103–129, 1995.