2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

PourNet: Robust Robotic Pouring Through Curriculum and Curiosity-based Reinforcement Learning

Edwin Babaians †‡, Tapan Sharma ‡, Mojtaba Karimi, Sahand Sharifzadeh*, and Eckehard Steinbach

***Abstract*— Pouring liquids accurately into containers is one of the most challenging tasks for robots as they are unaware of the complex fluid dynamics and the behavior of liquids when pouring. Therefore, it is not possible to formulate a generic pouring policy for real-time applications. In this paper, we propose PourNet, as a generalized solution to pouring different liquids into containers. PourNet is a hybrid planner that uses deep reinforcement learning, for end-effector planning, and Nonlinear Model Predictive Control, for joint planning. In this work, we introduce a novel simulation environment using Unity3D and NVIDIA-Flex to train our agents. By effective choice of the state space, action space and the reward functions, we allow for a direct sim-to-real transfer of the learned skills without additional training. In the simulation, PourNet outperforms state-of-the-art by an average of 4.9g deviation for water-like, and 9.2g deviation for honey-like liquids. In the real-world scenario using Kinova Movo Platform, PourNet achieves an average pouring deviation of 2.3g for dish soap when using a novel pouring container. The average pouring deviation measured for water was 5.5g. All comprehensive experiments and the simulation environment is available at: `http://cxdcxd.github.io/RRS/`.**

I. Introduction

Our ability to accurately pour liquids into containers might seem trivial to us. On the contrary, such an ordinary interaction is one of the most challenging tasks in Robotics. With automation entering more and more aspects of our professional and private life, pouring liquids is becoming an indispensable skill for robots. The potential of using precision pouring robots is immense in the catering/hospitality industry. Likewise, in service robotics, precision pouring as a skill will make robots valuable kitchen assistant robots. Furthermore, robots can aid as drinking and eating assistants for motor-impaired patients or the elderly.

Given the complexity of the fluid dynamics and the limited degrees of freedom in robots, formulating a direct control mechanism for pouring is difficult. For example, solving the Navier-Stokes [1] formulation to predict the behaviour of liquids in 3D is still an open millennium prize problem. Therefore, most of the recent works simplify the state space or the action space of the pouring problem. This results in failure under real world scenarios and to the best of our knowledge, there hasn't been any robust controller that can handle the robot pouring scenario using different liquids.

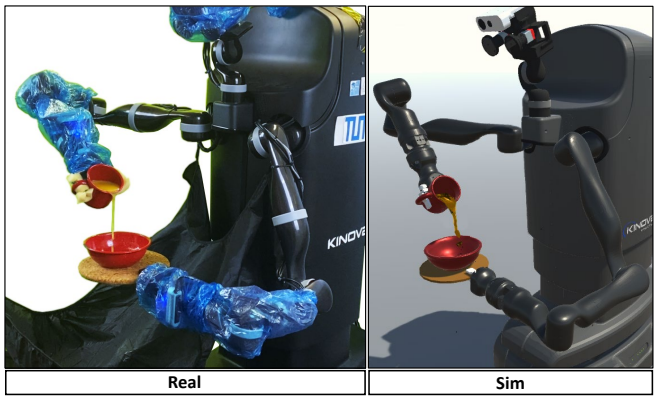

Fig. 1: On the right, the simulation environment is illustrated during the pouring of a water-like liquid. On the left, the robot is pouring orange juice in a real environment using our trained policy network, PourNet. PourNet learns to perform natural human-like pouring actions through several simulated experiences.

Recent progress in deep reinforcement learning (RL) [2] and inverse reinforcement learning (IRL) [3] has brought promising results for controlling agents in complex environments with large state or action spaces. This motivated us to solve the robotic pouring using deep reinforcement learning. One of the challenges in applying RL to robotics problems is that acquiring a large number of experiences from the real world can be time and cost expensive. Therefore, often it is more beneficial to carry out the training in a simulation environment and later transfer the skill to the real world. To this end, we implemented a simulation environment for the interaction between robots and the liquids. Through extensive real-world and simulated experiments, we design the optimized state-space, action-space, and the reward functions. As a result not only the robot can easily learn the tasks within the simulation environment, but the transition from simulation to real goes smoothly.

In summary, here are the main contributions of our paper:

- Implementing a simulation environment to model the interaction between robots and liquids, using Unity3D, ML-Agents, and Nvidia Flex and formulating robotic pouring within the RL framework. This includes using Curriculum Learning for liquid parameters randomization. By increasing the difficulty as the curriculum advances, the robot learns a pouring policy that can be generalized to various liquid types.
- Proposing the first hybrid arm controller that combines an RL-based approach for planning the end-effector

All authors are with the Department of Electrical and Computer Engineering, Chair of Media Technology (LMT) and Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM), and * Ludwig Maximilian University of Munich (LMU), Germany. - {edwin.babaians, tapan.sharma, mojtaba.karimi, eckehard.steinbach}@tum.de and {sharifzadeh}@dbs.ifi.lmu.de

This work has received funding from the Bavarian Ministry of Economic Affairs, Regional Development and Energy as part of the project '5G Testbed Bayern mit Schwerpunktanwendung eHealth'.

† Corresponding author
‡ Equally contributing authors

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981195

TABLE I: PourNet compared to the other state-of-the-art approaches

Evaluation Criteria	Closed-loop Control [4]	LfD [5]	MPNet [6]	Vision CNN [7]	DDPG [8]	**PourNet**
Liquid Properties Awareness	✗	✗	✗	✗	✓	✓
Container Geometry Awareness	✓	✗	✗	✗	✗	✓
Ability to control multiple actions	✗	✓	✓	✗	✓	✓
Generalization to novel liquids	✗	✗	✗	✗	✓	✓

with an NMPC-based [9] solver (Nonlinear Model Predictive Control) to exploit the maximum degrees of freedom and therefore generating smooth transitions of the arm.

- Successfully pouring several liquids with different viscosities to a container in real world by directly using the skills that are learned in a simulation and without any additional fine-tuning in the real-world.

The rest of the paper is organized as follows. In Section II, we briefly overview the state-of-art approaches. In Section III, we introduce the problem statement. In the Section IV, the proposed system architecture and all the technical details are discussed. Our experiments are presented in the Section V. Finally, the Section VI concludes the paper and discusses the future work.

II. Related Works

Table I summarizes the state-of-the-art solutions and compares them with PourNet on different evaluation criteria. Based on the related work for precision pouring, the approaches can be classified into two categories:

A. Closed-loop Control Mechanisms

Closed-loop control approaches do not require data-driven methods and use mechanisms such as Proportional-Integral-Derivative (PID) controllers. For instance, Dong et al. proposed a PD controller-based approach [4]. In their approach, they propose calculating the volume of the poured liquid using the model of the target container and the height of liquid contained in it. Likewise, for controlling actions of pouring, they propose using a (PD) controller, using pouring container's angular speed as a process variable and poured volume as a control variable. A prominent drawback observed was the inability to factor in the effect of surface tension of the water. Since the PD-based mechanism is not aware of these liquid-specific variables, it cannot generalize to get the best results.

B. Machine Learning Approaches

1) ***Learning from Demonstrations (LfD):*** LfD is one of the popular methods to teach robots a particular skill [3], [10], [11]. Usually, a human teacher, demonstrates the skill using teleoperation that a robot can mimic to perform. Rozo et. al proposed an innovative state-of-the-art approach using LfD and force feedback-based perception [5]. The underlying policy is based on a Parametric Hidden Markov Model for Behavior Cloning. However, the model, is blind towards the type of the liquid it works with. As a result, for each novel liquid, a new set of demonstrations would be required and hence is not generic for all liquid types.

2) ***Haptics and Audition:*** In [6] Linag et al. propose a Multimodal Pouring network (MP-Net) to robustly predict the liquid height by considering audio and haptics as input. Using raw audio data, a spectrogram with 257 descriptors is created. The haptic data associated with each time slice is gradually fed into the encoder module (a recurrent neural unit). In the final step, the height predictor module calculates the 1D length of the air column in the target container. Using a multimodal pouring data set including 300 recordings and force-torque measurements, the MP-Net is trained based on three types of containers. However, it suffers from the same prominent drawback, where the underlying model is blind towards the liquid properties. This is observed as a poor generalization of MP-Net to liquids of approximate similar nature like milk and fruit juices [6] which perform poorly, although for water the results are very promising.

3) ***Visual Feedback:*** Schenck et al. [12] described a machine-vision-based approach which uses a visual feedback to perform closed-loop control for pouring liquids [7]. To detect liquids, convolutional neural networks (CNNs) are used to detect the type of container and the volume of liquid within it based on RGB and thermal images [13]. The CNN-based output is restricted to merely 10 classes of volume. This approach hence, is not completely representative for precise pouring. Based on the pouring model trained on a sequence of 279 pouring actions, the pouring policy tested on water, observed an average 38ml deviation [7] from the target amount. Besides, the model is unaware of the liquid properties. In addition, the scenarios involving occlusion of camera feedback would greatly impact the performance.

4) ***Reinforcement Learning:*** Chau et. al [8] presented a deep reinforcement learning-based approach to learn a policy for pouring using Deep Deterministic Policy Gradients (DDPG) [14]. In their work, the training experiences were collected using a state-of-the-art liquid simulator PreonLab [15]. Through their experiments, performed with a PR2 robot, they demonstrated a successfully transfer of the learned policy to a real robot. Despite the absence of force sensors, the task involved pouring liquids using a robot at specific heights while avoiding spills and collisions. Using force/torque sensors can on contrary provide means for better precision pouring control.

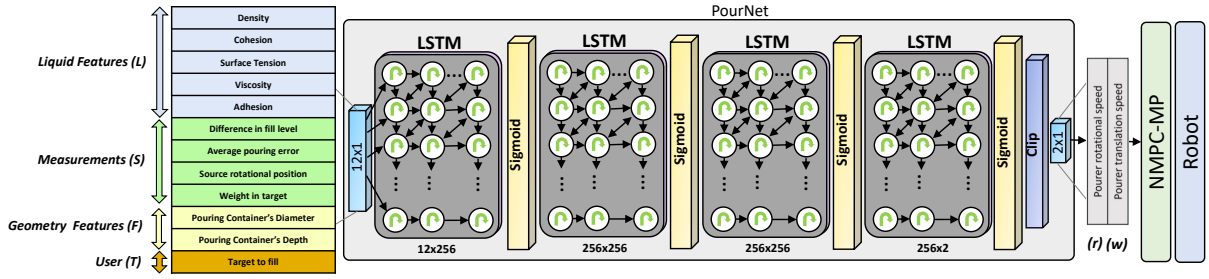

Fig. 2: PourNet Model Architecture.

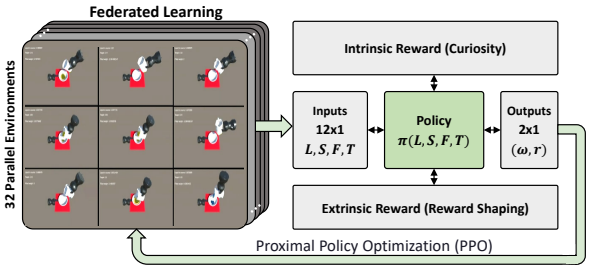

Fig. 3: Training Architecture.

III. Problem Statement

PourNet is a continuous control RL model for liquid pouring. Given a set of observations in an operating environment, PourNet generates actions from a policy that is trained to maximize its cumulative reward (Figure 3). This can be described as:

Set of observations:

L: Set of liquid features,

S: Set of state information,

F: Feedback measurements from the environment,

T: Desired liquid to be poured.

Set of actions:

ω: Pouring Container's rotational velocity,

r: Pouring Container's translational velocity.

Such that,

$$(\omega, r) = \pi(L, S, F, T) \tag{1}$$

Where, π is the policy and *Success Criteria* will be satisfied if the poured quantity is within some tolerable deviation of T.

IV. Technical Details on PourNet

The success criteria described in Section III makes precision pouring a "Sparse Reward Reinforcement Learning" problem. This is attributed to the fact that the agent can only receive reinforcing rewards from the environment, if the poured quantity in the receiving container is within some tolerable deviation. From the technical point of view, this paper proposes *PourNet* as an intelligent agent which is capable of pouring a liquid with liquid property awareness, container geometry awareness and environment awareness using force/torque based perception.

A. Average Pouring Error

We introduce "**Avg. Pouring Error**" as a performance metric. This is defined as the average pouring deviation over n successfully completed pouring episodes. We use Avg. Pouring Error to evaluate the agent's progress during training. Therefore, we design the extrinsic rewards of an agent as a function of Avg. Pouring Error such that the agents with a tolerable average pouring errors are rewarded additionally to the consequence of the actions taken during a single episode. This allows us to also use this metric for hyperparameter tuning such that the average pouring error does not increase many-fold over time.

B. Proximal Policy Optimization

The RL algorithm considered in this paper is based on Proximal Policy Optimization (PPO) [16]. The choice of PPO was based on a training experiment in the simulation setup using a single liquid profile. For this experiment, it was observed that the PPO observed a better average pouring error when compared to the Soft-Actor-Critic (SAC) [17] based RL algorithm.

C. Action Handler

PourNet's policy described in Section III outputs two floating number values, i.e. translation and rotation speed; therefore, the action handler of the agent is designed to reflect these values in terms of 6D poses and 3D motions. In Algorithm 1, the paper discusses the action handler, which converts the PourNet's actions to the subsequent 3D motions. The training of PourNet is always **without** the real robot in the loop. However, for transferring the policy to a robot, the pouring container's pose essentially becomes the manipulator's end-effector. This is because the pouring container is held within the grippers, and hence, becomes the end-effector for the motion-planner to execute the actions. Ultimately, the Non-linear Model Predictive Control-based Motion Planner (NMPC-MP) [18] receives the final trajectory.

D. Reward Shaping

As established in the introduction of the Section IV, the PourNet's RL problem essentially is a Sparse Reward problem. This has a great implication on the agent to train for development of a generalized pouring policy, as it's not guaranteed that the agent will see positive reinforcing rewards in each episode of pouring. The extrinsic reward shaping in this context is described in Algorithm 2.

E. Intrinsic Curiosity Module

The reward function as discussed in algorithm 2, is essentially a Sparse Reward Problem. In the absence of recurrent reinforcing rewards, this means that an agent requires a large

number of the training steps to explore the environment for training a robust policy. One way to essentially improve this situation is to use an intrinsic reward-based system. In addition to the extrinsic rewards, intrinsic rewards can aid an agent to explore it's environment along with exploitation of the policy. This paper uses one such intrinsic reward-based system called the "Intrinsic Curiosity Module" (ICM) based on the work of Pathak et al. [19]. The idea of curiosity-driven learning is to build a reward function intrinsic to an agent, such that it acts as a self-learner, and improves it's own policy based on the intrinsic reward. This involves using the feature representation of the states of two successive time steps to predict the current time-step's actions and using current time step's actions and feature representation of the states to predict next time step's feature representation of the state. The discrepancy in prediction of next feature representation of the states, finally forms the basis of the intrinsic reward signal. A large prediction error is indicative of exploration of uncharted environment. This in turn helps the agent to explore, thus, finding better extrinsic results in the future. In this paper, a prominent observation is made in the PourNet's performance improvement when using the ICM. Figure 5, showcases this improvement for LSTM-based PourNet. As compared to the standalone PPO-based model, using ICM sees a steady improvement of the performance in terms of reducing average pouring errors.

F. Curriculum Learning

Curriculum Learning [20] is inspired from human behavior of breaking down a complex difficult task into smaller structured tasks with increasing difficulty.

As an inherent part of humans learning from their environment, it can be observed that learning is certainly better and more efficient if it's organized into some meaningful order. The order is illustrative of gradually increasing concepts, and complexity. Hence, an intelligent agent using curriculum learning approach, first starts out with only easy examples of a task and then gradually increases the task difficulty.

This is often considered as a robust Environment Parameter Randomization strategy divided into a hierarchical structure of lessons. An agent tries to solve a task using easier lessons at first. As the policy becomes good in determining appropriate actions for the observations, gradually, the difficulty of the next lessons is increased.

PourNet uses the Curriculum Learning as a primary "**Liquid Properties Randomization Strategy**". Some liquids are easier to handle for pouring. However, as viscosity and adhesion in liquids tend to increase, the pouring task becomes more challenging. As a result, PourNet uses a hierarchically designed curriculum for liquid parameter randomization.

1) Designing Pouring Curriculum: This paper considers the following liquid properties for designing a curriculum-based pouring scenario: Density, Rest Particle Distance, Adhesion, Cohesion, Surface Tension, Viscosity.

As such, out of these enumerated liquid properties, the effect of adhesion, cohesion, surface tension, and viscosity affects the pouring behavior to a greater extent. These are either sources of spillage, or factor in the Naiver-Stokes [1] formulation describing fluids in motion. For instance, more viscous and adhesive liquids like honey and ketchup are difficult to pour as compared to the less viscous ones like water. As a result, the curriculum can be designed with these four properties in order of increasing values, representative of the increasing complexity of the task. Density and Rest Particle Distance, on the other hand, can be domain randomized using a simple uniform sampling strategy.

The PourNet's curriculum as proposed in this paper in terms of randomized Nvidia Flex liquid parameters has been depicted in the Figure 4. The PourNet's performance improvement can be seen in the Figure 6 when using Cur-

Algorithm 1: Action handler algorithm uses output from PourNet to plan 3D motion at Robot's end effector.

Given:
$\vec{S} \leftarrow$ *Pouring container's Position.*
$\vec{T} \leftarrow$ *Receiving container's Position.*
$L_w \leftarrow$ *Weight of liquid in the receiving container.*
$W_r \leftarrow$ *Target to be filled in the receiving container.*
$\Delta L_w \leftarrow$ *Tolerance in deviation.*
$T_{geometry} \leftarrow$ *Geometry of receiving container defined by center* $\vec{T}$ *and extents.*
Input: (ω, r) Output from PourNet
Output: (Δx, $\Delta\theta$) 3D Motion at the robot's end effector

```
if S_x ∉ [T_geometry_xmin, T_geometry_xmax] & S_z ∉ [T_geometry_zmin, T_geometry_zmax] then
    if S_y − T_y ⩾ 2 × T_geometry_yextents + 0.05 then
        d⃗ = T⃗ − S⃗
    else
        d⃗ = (T_x, S_y, T_z) − (S_x, S_y, S_z)
            // d⃗ ← Direction of Motion.
    Δx = S⃗ + (r × Δt) × d⃗
    output Δx
else
    if L_w ⩽ (W_r ± ΔL_w) then
        Δθ = ω × Δt ∀ω ⩾ 0
    else
        Δθ = ω × Δt ∀ω < 0
            // Δθ ← Rotation about pouring container's forward axis.
    output Δθ
```

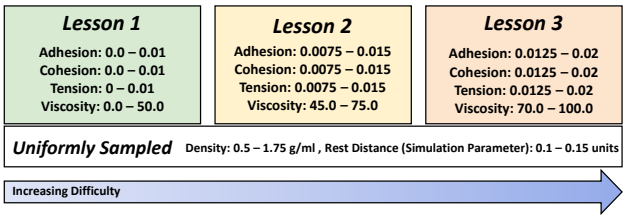

Fig. 4: Curriculum Learning lessons and parameters. The idea is to domain randomize the parameters in increasing order of their difficulty in pouring dynamics.

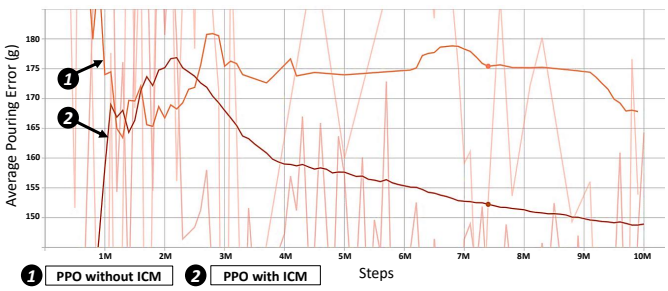

Fig. 5: Performance Improvement by using ICM. Smoothed lines 1 and 2 are displayed on the graph, and raw data is shown in the background. This is related to simulation.

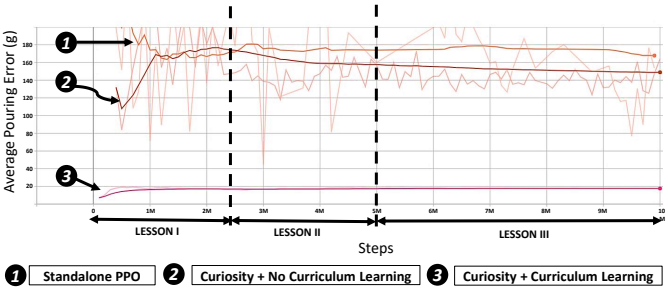

Fig. 6: Performance Improvement by using Curriculum Learning. This is related to simulation.

riculum Learning along with curiosity. This is compared to the scenario where PPO is used standalone and where PPO is used along with the ICM. This conforms with the expectation as the policy is trained with a three-lesson curriculum of increasing difficulty. Hence, policy performs better from the first lesson itself as it sees enough positive rewards from the environment and adjusts it's knowledge as the difficulty of the lesson begins to increase.

G. End-to-end Technical Specifications

The details of the overall technical specifications, for PourNet as seen in Figure 3 are:

1) **PourNet**: It is the intelligent agent in the environment which observes the input and takes action by using it's knowledge from the pouring policy.
2) **Observations**: The observation is a 12x1 vector with the following inputs. *Liquid Features*: The liquid features are a 5x1 vector of observation including information of Density, Cohesion, Surface Tension, Viscosity, and Adhesion. *Pouring Container Geometry*: It is a 2x1 vector of the pouring container's diameter and depth. *Force/Torque Feedback*: This includes a 1x1 vector of weight information of the liquid in the target container. *User Input*: This is a 1x1 input of the target level of the liquid to fill. *Measurements*: This is a 3x1 vector which includes measured observation from the environment and these include the difference in the fill level in the target container, average pouring error for all pouring trials until the current time step and the rotational tilt of the pouring container.
3) **ICM**: This is the Intrinsic Curiosity Module, which form the basis of rewards intrinsic to the PourNet.

Algorithm 2: PourNet: Extrinsic Reward Shaping Algorithm

Given:
$Coll$: *State of collision between pouring and receiving container.*
ϵ_t: *Current episode's pouring error.*
$\Delta\epsilon$: *Average Pouring error over past n successful episodes.*
$L_w \leftarrow$ *Weight of liquid in the receiving container.*
$W_r \leftarrow$ *Target to be filled in the receiving container.*
$\Delta L_w \leftarrow$ *Tolerance in deviation.*
Input: $(\epsilon_t, \Delta\epsilon)$
Output: r_t^e Extrinsic reward for the current episode.
$r_t^e = 0$
while $L_w \leqslant (W_r \pm \Delta L_w)$ **do**
  $r_t^e = r_t^e + \frac{-1}{Max.\,Number\,of\,Steps}$
  if $Coll\,is\,True$ **then**
    $r_t^e = r_t^e - 1$
    return r_t^e
if $\epsilon \leqslant 10\,grams$ **then**
  $r_t^e = r_t^e + 1$
  if $\Delta\epsilon \leqslant 20\,grams$ **then**
    $r_t^e = r_t^e + 1$
output r_t^e

4) **Actions**: The PourNet takes actions based on the observation by exploiting it's underlying pouring policy which updates the pouring container's rotational and translational positions in the scene.
5) **Environment**: It is the simulation environment where the PourNet gains the experiences for a robust policy.

H. PourNet's Architecture

Overall, two contesting deep neural network architectures were considered for the PourNet. One based on a recurrent neural network using LSTM [21] as foundational blocks and the other using a feed-forward neural network.

The primary investigation of this paper is to establish the applicability of the RL to precision pouring problem. Hence, the aforementioned architectures forms the basis of underlying model which uses RL for policy modeling.

1) LSTM-based PourNet: The LSTM-based PourNet architecture is described in Figure 2. The architecture has the following features: Number of Layers: 3, Number of hidden units: 256 units per hidden layer, Memory Size: 256, Sequence Length: 64.

2) Feed-forward neural network-based PourNet: Another architecture for the PourNet is based on the Feed-forward neural network (FNN). The PourNet's FNN-based architecture is similar to the LSTM-based one, where instead of a LSTM block, a feedforward neural network is used. The performance comparison is as described in Table IV.

V. Experimental Evaluation

A. Simulation

1) Simulation Setup: The primary precision pouring setup discussed in this paper is based on a novel simulation

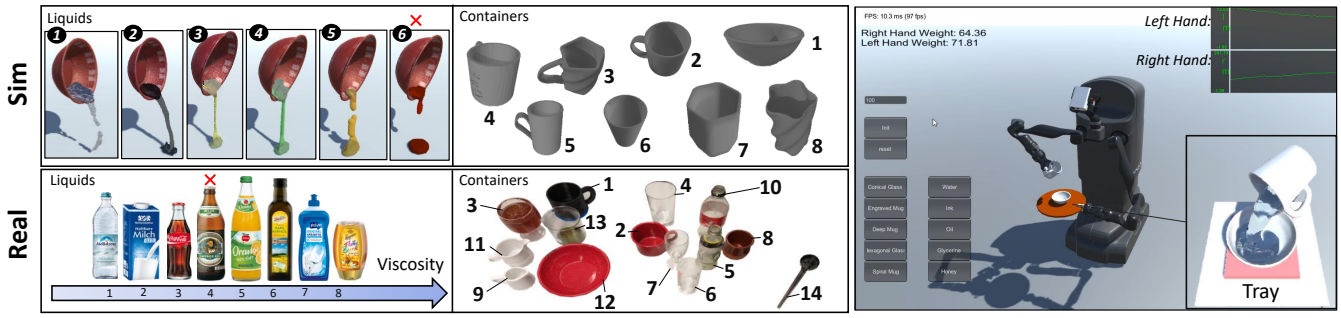

Fig. 7: The simulated (top) vs real (bottom) liquids and containers. On the right the test environment with robot in the loop is illustrated.

environment. The basis of our simulation environment are the Unity3D Game Engine [22] and the NVIDIA® Flex a *Position Based Fluid* [23] engine, and NVIDIA® PhysX 4.0 for rigid body simulations. The Machine Learning (ML) in the Unity3D® is catered to by ML-Agents [22] library. The overall end-to-end pipeline is as described in Figure 3. We tested on the simulated robot with a motion planner in the loop by bridging it to ROS via the approach proposed in [24]. We used eight 3D scanned containers which were available.

2) Training: The PourNet-based policy was trained for maximum *10 million time-steps* in a **Federated Learning** approach [25]. There are 32 parallel environments that independently provide a set of observations and actions to collaboratively train a single brain i.e. PourNet-based pouring policy. Essentially, each environment keeps all the local training data private to itself. By using this approach, not only is the training sped up, but the experiences are randomized as well. Randomness is a key to robust policy learning. The curriculum learning was applied with the lesson progression as 25% of time-steps for the first lesson, further 25% for the second lesson, and finally, remaining 50% of the time-steps for the third and the final lesson. Training and simulations was done using two NVIDIA® Quadro RTX6000 GPUs.

3) Discussion: Table IV enumerates the results of the average pouring error for the simulated liquid profiles as shown in Figure 7 with the NVIDIA Flex parameters as enlisted in Table III. Each simulated liquid profile was poured 10 times using a trained PourNet in the pure simulation environment using a randomized pouring container as shown in Figure 7. It can be observed that when comparing the LSTM-based PourNet with FNN-based PourNet, LSTM-based PourNet performs slightly better than the FNN-based PourNet for all simulated liquid profiles. Because of this better performance, all further experiments and results in the subsequent sections are using LSTM-based PourNet. There is a 26.7% improvement for water, 16.6% for ink, 11.94% for oil, 15.6% for the glycerine, 4.5% for honey, and finally 8.04% for the ketchup in terms of average pouring deviation. It should be noted that, since the simulated ketchup profile is very adhesive and also viscous, in absence of shaking action from the PourNet, the higher targets could not be filled.

B. Real Experiments

1) Platform Setup: The experimental setup to transfer PourNet to a real robot is based on the Kinova® Movo. As such, the experimental setup includes the following sub-systems: Robotiq HandE Gripper, BotaSys® SENSONE force/torque sensor operating at 100 Hz, and Intel® RealSense D435i only as Inertial Measurement Unit (IMU) at 200 Hz. Figure 8 highlights the overall experimental setup with the appropriate modules labeled. In all of our experiments, the stationary left arm is holding a tray with a radius of 10cm, and the pouring is carried out by the right arm.

2) Weight Measurement: For accurate object in hand weight measurements, we should eliminate the effect of the gripper's weight. After applying the gravity compensation using the IMU sensor [26] by factoring in the sensor bias, an accurate reading can be provided. Additionally, as a result, the sensor possesses a noise-free resolution of as low as 0.005 Nm torque for stationary left hand and 0.01 Nm torque for moving right hand at the 100 Hz operating frequency. In our experiments we are using the torque measurements on the left arm to measure the weight of the target container. However, in order to ensure a smooth signal from the sensor, filtering must be performed in order to safeguard PourNet's observations from short-duration spikes. This might result in unpredictable action response, which might be hazardous. As a result, for filtering the raw force/torque reading from the sensors, a second-order *Butterworth Low-Pass Filter* [27] was designed. On the left arm, we achieved an accuracy of 2 grams for the target weight measurements.

3) Experimental Containers & Liquids: Figure 7 describes the containers and liquids used for pouring. As such, ten novel pouring containers were used for the experimental evaluation along with one 3D printed training container. These cover a wide range of container types demonstrating variation in geometrical features. Likewise, seven liquids were considered for the pouring experiments. These liquids cover a wide range of experimental liquid features demonstrating generalized behavior of the PourNet.

4) Pouring Experiments: Figure 10 describes the action sequence that the robot follows for precision pouring task. The pouring experiments were divided into two categories i.e. *I*: demonstrating the generalization to different liquid types, and *II*: demonstrating generalization to different container geometries. In the experiment *I*, the pouring container was kept the same. The results for the experiment *I* has been enumerated in the Table V. For each liquid, the container 2

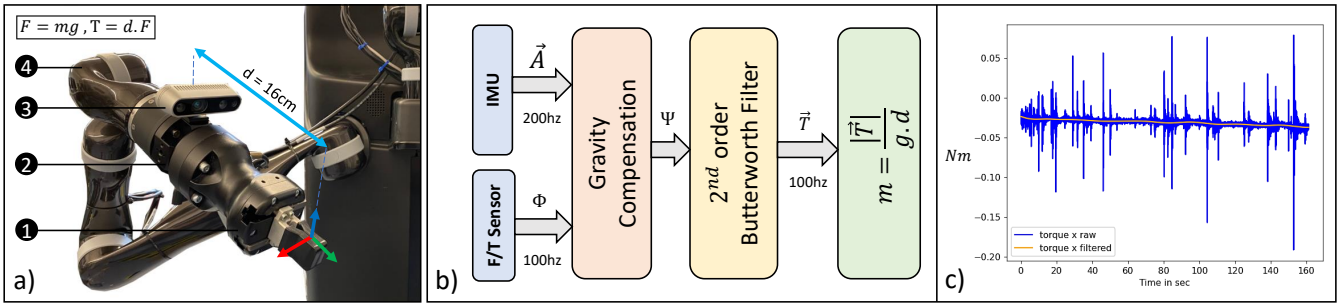

Fig. 8: a) Experimental Setup with Kinova Movo. b) Force measurement steps using F/T sensor. c) Second-order ButterWorth Low-pass Filter, operating at 100 Hz, with a cut-off frequency of 1 Hz.

TABLE II: Average Pouring deviation for water compared to the state-of-the-art approaches

Liquid	Closed-loop Control [4]	MPNet [6]	MultiFrame CNN (thermal vision) [7]	DDPG [8]	**PourNet**
Water	0.95±1.16 ml to 11.88±4.63 ml	6.3±6.1 ml to 23.7±18.7 ml	38 ml	19.96 ml	**3.8 ± 0.69 ml** to **8.66 ± 1.52 ml**

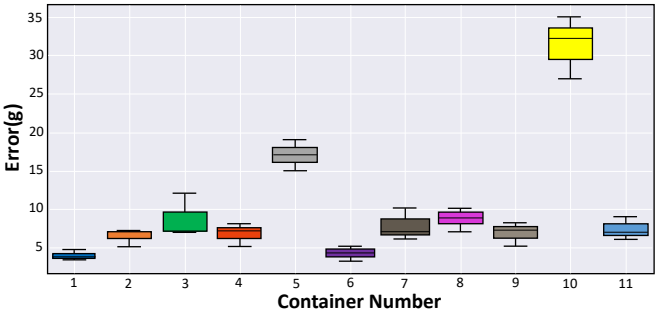

Fig. 9: Experiments using water and different containers.

TABLE III: Simulation Liquid Profiles (Flex Parameters)

Liquid	Adhesion	Cohesion	Surface Tension	Viscosity
Water	0.0001	0.001	0.005	0.01
Ink	0.0001	0.0025	0.0075	0.05
Oil	0.001	0.001	0.0001	6.5
Glycerine	0.005	0.001	0.0001	50
Honey	0.025	0.001	0.0001	65
Ketchup	0.1	0.001	0.00001	80

TABLE IV: LSTM-based PourNet: Experiment with liquid simulation profiles

Type	Target(g)	Water	Ink	Oil	Glycerine	Honey	Ketchup
LSTM	50	**7.72**	**4.44**	**4.66**	**4.4**	**9.1**	**11.66**
LSTM	100	**2.3**	**6.34**	**8.64**	**7.66**	**5.26**	20.36
LSTM	150	**4.96**	**8.38**	**6.44**	**8.8**	12	-
LSTM	200	**4.76**	**5.9**	**7.38**	**6.9**	**10.54**	-
FNN	50	8.49	7.65	5.35	8.3	9.4	15.42
FNN	100	5.13	7.94	8.91	8.52	8.26	**19.40**
FNN	150	7.65	8.32	8.3	8.6	**10.2**	-
FNN	200	5.65	6.17	8.25	7.5	10.79	-

as seen in the Figure 7, was used to pour three target profiles i.e. 50g, 100g, and 150g. Against each target, a liquid was poured thrice. Likewise, for the experiment *II*, the liquid type was fixed by choosing water for each pouring container. For each container, 50g of water was poured thrice. The Figure 9 depicts the pouring experiment for the experiment *II*. Beer cannot be simulated due to its effervescence and subsequent large foam formation, therefore in the real experiments it failed (see Figure 10) .

TABLE V: Deviation from target using different liquids

Target (g)	Water	Milk	Cola	Orange Juice	Rapeseed Oil	Dish Soap	Honey
50	6.33	3.33	6.00	2.67	8.55	3.33	9.33
100	3.55	3.66	7.55	2.33	5.67	1.67	8.66
150	7.33	5.53	6.00	4.67	8.55	2.12	5.23

5) Discussion: From the Figure 9, it can be observed that the errors are within 10 grams for most of the pouring containers. It can be noted however, that container N.5 and container N.10 are outliers. This can be attributed to the fact that these are bottles, and, were not a part of training setup. As a result, PourNet had no experience in dealing with such objects while training. Hence, ignoring the outliers, on comparing the performance for pouring water with some of the state-of-the-art methodologies, the results are as enlisted in Table II. It can be observed that the PourNet outperforms these with small errors.

VI. Conclusions

This paper explored a RL method for precision pouring problem. Precision pouring was primarily driven by curiosity and curriculum-based reinforcement learning. Based on the trained PourNet, various experiments were carried out in the both simulation and by transferring the trained policy on a real robot. Based on the experiments, we observed generalization of the PourNet to different liquid types. The performance in terms of pouring deviations ranged from 2.33 g for dish soap to 7.55 g for honey. Likewise, the paper demonstrated PourNet's ability to generalize to novel pouring containers. The performance varied from 3.8±0.69 ml for

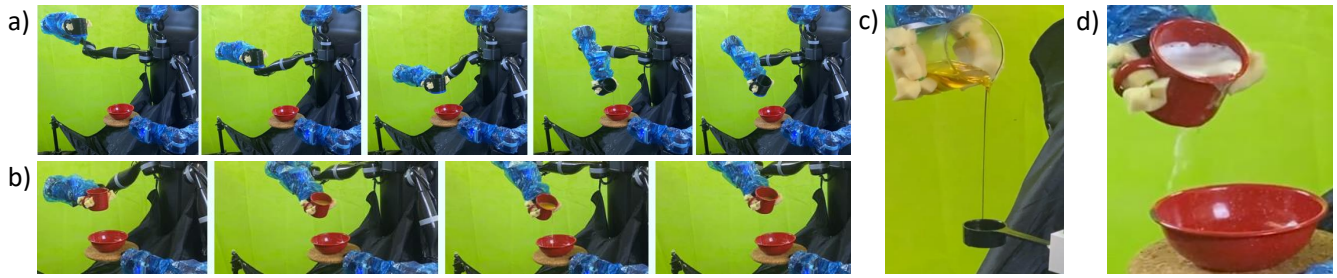

Fig. 10: a) Denotes starting point generalization. b) Denotes the juice pouring task with container (N.2 to N.12). c) Denotes the oil to the novel spoon pouring experiment, container (N.13 to N.14) d) Denotes the beer pouring failure.

the best performing container to the 8.6 $\pm$1.52 ml for the worst performing container. This thus demonstrates PourNet as a precise pouring agent with an ability to generalize for different liquid types and different pouring container geometries. One of the interesting directions for future works is exploring other intrinsic reward signals using techniques such as Generative Adversarial Imitation Learning (GAIL) [28] as well as including more extensive visual signals such as scene graphs [29], [30].

REFERENCES

[1] P. Constantin and C. Foias, *Navier-stokes equations*. University of Chicago Press, 1988.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," *arXiv preprint arXiv:1612.03653*, 2016.

[4] C. Dong, M. Takizawa, S. Kudoh, and T. Suehiro, "Precision pouring into unknown containers by service robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5875–5882.

[5] L. Rozo, P. Jiménez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in *9th International Workshop on Robot Motion and Control*. IEEE, 2013, pp. 227–232.

[6] H. Liang, C. Zhou, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun, M. Stoffel, and J. Zhang, "Robust robotic pouring using audition and haptics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 880–10 887.

[7] C. Schenck and D. Fox, "Visual closed-loop control for pouring liquids," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2629–2636.

[8] C. Do, C. Gordillo, and W. Burgard, "Learning to pour using deep deterministic policy gradients," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3074–3079.

[9] S. Hu, "Nonlinear model predictive control based real-time motion planning for robot manipulator," 03 2021.

[10] Y.-Y. Tsai, B. Xiao, E. Johns, and G.-Z. Yang, "Constrained-space optimization and reinforcement learning for complex tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 683–690, 2020.

[11] M. Akbulut, E. Oztop, M. Y. Seker, X. Hh, A. Tekden, and E. Ugur, "Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing," in *Conference on Robot Learning*. PMLR, 2021, pp. 1896–1907.

[12] M. Kennedy, K. Schmeckpeper, D. Thakur, C. Jiang, V. Kumar, and K. Daniilidis, "Autonomous precision pouring from unknown containers," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2317–2324, 2019.

[13] R. Mottaghi, C. Schenck, D. Fox, and A. Farhadi, "See the glass half full: Reasoning about liquid containers, their volume and content," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1871–1880.

[14] 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," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[15] "Fifty2 preonlab - avl.com," https://www.avl.com/fifty2-preonlab, (Accessed on 03/01/2022).

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] 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*. PMLR, 2018, pp. 1861–1870.

[18] S. Hu, E. Babaians, M. Karimi, and E. Steinbach, "Nmpc-mp: Real-time nonlinear model predictive control for safe motion planning in manipulator teleoperation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 8309–8316.

[19] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.

[20] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar *et al.*, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[23] M. Macklin and M. Müller, "Position based fluids," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–12, 2013.

[24] E. Babaians, M. Tamiz, Y. Sarfi, A. Mogoei, and E. Mehrabi, "Ros2unity3d; high-performance plugin to interface ros with unity3d engine," in *2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*. IEEE, 2018, pp. 59–64.

[25] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[26] D. Kubus, T. Kroger, and F. M. Wahl, "On-line rigid object recognition and pose estimation based on inertial parameters," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1402–1408.

[27] S. Butterworth *et al.*, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[28] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, pp. 4565–4573, 2016.

[29] S. Sharifzadeh, S. M. Baharlou, and V. Tresp, "Classification by attention: Scene graph classification with prior knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 6, 2021, pp. 5025–5033.

[30] S. Sharifzadeh, S. M. Baharlou, M. Schmitt, H. Schütze, and V. Tresp, "Improving scene graph classification by exploiting knowledge from texts," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2189–2197.