Parallel Droplet Control in MEDA Biochips Using Multi-Agent Reinforcement Learning

Tung-Che Liang [1] **Jin Zhou** [1] **Yun-Sheng Chan** [2] **Tsung-Yi Ho** [3] **Krishnendu Chakrabarty** [1] **Chen-Yi Lee** [2]

Abstract

Microfluidic biochips are being utilized for clinical diagnostics, including COVID-19 testing, because they provide sample-to-result turnaround at low cost. Recently, microelectrode-dot-array (MEDA) biochips have been proposed to advance microfluidics technology. A MEDA biochip manipulates droplets of nano/picoliter volumes to automatically execute biochemical protocols. During bioassay execution, droplets are transported in parallel to achieve high-throughput outcomes. However, a major concern associated with the use of MEDA biochips is microelectrode degradation over time. Recent work has shown that formulating droplet transportation as a reinforcement-learning (RL) problem enables the training of policies to capture the underlying health conditions of microelectrodes and ensure reliable fluidic operations. However, the above RL-based approach suffers from two key limitations: 1) it cannot be used for concurrent transportation of multiple droplets; 2) it requires the availability of CCD cameras for monitoring droplet movement. To overcome these problems, we present a multi-agent reinforcement learning (MARL) droplet-routing solution that can be used for various sizes of MEDA biochips with integrated sensors, and we demonstrate the reliable execution of a serial-dilution bioassay with the MARL droplet router on a fabricated MEDA biochip. To facilitate further research, we also present a simulation environment based on the PettingZoo Gym Interface for MARL-guided droplet-routing problems on MEDA biochips.

1. Introduction

In recent years, we have seen progress on the use of deep reinforcement learning (RL) to assist sequential decision-making problems, such as games (Silver et al., 2017; Vinyals et al., 2019; Brown & Sandholm, 2019), robotics (Gu et al., 2017), autonomous driving (Sallab et al., 2017; Palanisamy, 2020; Wachi, 2019), quantitative trading strategies (Lee et al., 2020), and healthcare systems (Manak et al., 2018; Liang et al., 2020a). Many of these successful RL applications involve more than one agent or player, which naturally leads to the setting of multi-agent RL (MARL). MARL addresses the decision-making problem for multiple agents in a common environment, where each agent's goal is to optimize its own long-term reward by interacting with the environment and other agents (Zhang et al., 2019). In this paper, we demonstrate the application of MARL for droplet control in microfluidic biochips based on the microelectrode dot-array (MEDA) platform. We show that because the health condition of a biochip dynamically changes over time, the modeling of parallel droplet control in MEDA biochips as an MARL problem can ensure reliable bioassay execution with high throughput.

1.1. MEDA Biochips

The rapid worldwide spread and impact of the COVID-19 virus has created an urgent need for reliable, accurate, and affordable testing on a massive scale. For example, the National Institutes of Health (NIH) has launched the Rapid Acceleration of Diagnostics (RADx) initiative to develop and implement technologies for COVID-19 testing (NIH, 2021a). One of the most promising technologies for realizing this goal is microfluidics. A microfluidic biochip manipulates tiny amounts of fluids to automatically execute biochemical protocols for point-of-care clinical diagnosis with high efficiency and fast sample-to-result turnaround (Sun et al., 2020; Ganguli et al., 2020; Sheridan, 2020). Because of these characteristics, the RADx initiative has awarded grants to several biomedical diagnostic companies to develop microfluidic technologies that could dramatically increase testing capacity and throughput (NIH, 2021b; Ouyang et al., 2020). Other applications of microfluidics include screening of newborn infants (Sista et al., 2020;

[1] Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA [2] Department of Electronics Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan [3] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Correspondence to: Tung-Che Liang <tung.che.liang@duke.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

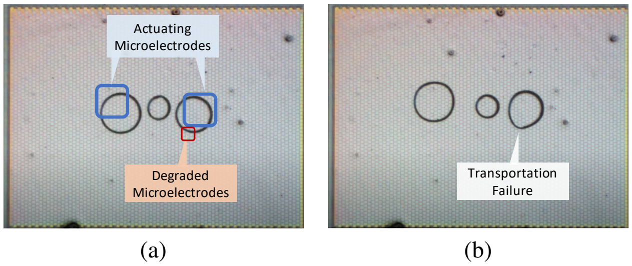
Inc., 2021), drug discovery (Li et al., 2020), and clinical diagnostics (Chou et al., 2015; Schmitz & Tang, 2018).

The MEDA biochip platform has been proposed in recent years to further advance microfluidics technology (Lai et al., 2015; Ho et al., 2016). A MEDA biochip is composed of a two-dimensional microelectrode array that manipulates discrete fluid droplets. MEDA biochips manipulate nanoliter droplets using the principle of *electrowetting-on-dielectric* (EWOD) (Pollack et al., 2000). When driven by a sequence of control voltages, the microelectrode array can perform fluidic operations, such as dispensing, mixing, and splitting (Wang et al., 2011; Zhong et al., 2020). Using MEDA biochips, bioassay protocols are scaled down to droplet size and executed through software-based control of nanoliter droplets. [1]

As microfluidic biochips are being used for critical point-of-care diagnostics, reliability in these systems has become an important focus of research (Zhong et al., 2020; Liang et al., 2020b). It has been reported that the unit cells (i.e., electrodes) of an EWOD biochip degrade over time (Verheijen & Prins, 1999; Su et al., 2006; Xu & Chakrabarty, 2007a; Drygiannakis et al., 2009). Electrode degradation results from charge trapping in the dielectric insulator (Dong et al., 2015); therefore, a degraded electrode cannot be observed using a CCD camera. MEDA biochips, in particular, are more susceptible to microelectrode degradation than other EWOD biochips. This is because microelectrodes in MEDA biochips are charged during not only droplet actuation, but also during droplet sensing, i.e., a microelectrode in MEDA biochips is charged more frequently than in other EWOD biochips (Zhong et al., 2020). An example of microelectrode degradation is shown in Figure 1 for a fabricated MEDA biochip. Three droplets are present on the biochip, and one of them is present over a group of degraded microelectrode. In the next time slot, two groups of microelectrodes are actuated to move two droplets. However, one of the two fluidic operations fails because unwanted surface-tension force is exerted by the degraded microelectrodes.

1.2. Motivation

To carry out a bioassay protocol with specified fluidic operations on a MEDA biochip, a synthesis tool is used generate a schedule of fluidic operations (Chakrabarty et al., 2010; Zhong et al., 2018). The operations are then mapped to on-chip modules to perform the operations. Next, the resultant droplet of one operation is used for the following operation, and thus the droplet needs to be transported from the previous module to the next. The problem of choosing the microelectrode path associated with droplet transportation and droplet-mixing pathway is referred to as *droplet routing*. Many synthesis methods have been proposed to accomplish high-throughput bioassay outcomes (Xu & Chakrabarty, 2007b; Keszocze et al., 2017; Zhong et al., 2018); these methods execute parallel droplet routing in a limited microelectrode-array space. However, because microelectrode degradation is not observable using a CCD camera, these methods cannot ensure reliable droplet transportation if the microelectrodes associated with the routing path degrade over time.



(a) (b)

Figure 1. Droplet transportation fails because of microelectrode degradation. (a) Three droplets are present on the electrode array. Two groups of microelectrodes are actuated to move two droplets. (b) After microelectrode actuation, the right droplet cannot be moved to the desired location completely because it was present over some degraded microelectrode; the left droplet is transported to the desired location correctly.

A recent study showed that formulating droplet transportation as a single-agent reinforcement learning (RL) problem enables the training of deep neural network policies to capture the underlying health conditions of the biochip and provide dynamic adaptation based on sensing results and reliable fluidic operations (Liang et al., 2020a). However, the above RL-based solution suffers from two key limitations. First, the RL framework does not consider the practical scenario of parallel droplet transportation, i.e., it assumes that the environment is *stationary*. During the execution of a typical bioassay, it is often the case that multiple droplets must be transported concurrently. However, the work in (Liang et al., 2020a) modeled the droplet-transportation task as a single-agent RL problem, and each agent acts on only one droplet. Therefore, without knowing the routing strategies of the other agents, the movement of one droplet can hinder the routing of another droplet. An even more severe problem is that, without "collaboratively" interacting with the other droplet-routing agents, droplets can collide with each other, resulting in bio-sample contamination and erroneous bioassay outcomes. Consider the example shown in Figure 2; two droplets need to be transported concurrently to the destinations. Because the agents that act on the two droplets do not know (or did not learn) the subsequent actions from the other agent, the two droplets can be contaminated during transportation.

Another limitation of (Liang et al., 2020a) is that this framework requires the integration of a CCD camera in the mi-

[1] A detailed description of MEDA biochips can be found in the supplementary document.

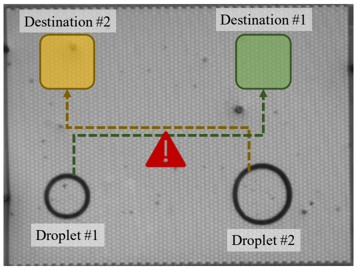


Figure 2. Two droplets need to be transported to their corresponding destinations on a DMFB. The solution in (Liang et al., 2020a) will cause the droplets to be contaminated on their way to the destinations.

crofluidic system to capture droplet locations in real time. However, as microfluidic biochips are developed for point-of-care and affordable clinical diagnostics, a portable microfluidic system may not allow for the integration of a CCD camera for real-time sensing.

To address the above limitations, we present an MARL framework for the parallel droplet-routing problem. The framework assigns a droplet routing task to an RL agent, and all the agents are modeled in a cooperative setting so that they can accomplish routing tasks without blocking each other. We also demonstrate the proposed MARL framework on a fabricated MEDA biochip, where real-time sensing is available using CMOS electronics integrated under each microelectrode (Lai et al., 2015).

1.3. Paper Contributions

This work is the first attempt to apply MARL to emerging microfluidic systems. The key contributions of this paper are as follows.

- We formulate a novel framework for MARL-based droplet routing on MEDA biochips. We discuss the challenges involved with the formulation of parallel droplet routing as an MARL task. In our framework, the policy is first trained in a simulated MEDA environment. The pre-trained policy is then loaded on the controller associated with a MEDA biochip, and the policy generates real-time droplet routing pathways.

- We demonstrate the routing scheme by executing a bioprotocol on a fabricated MEDA biochip. We show that the policy enables agents to learn to cooperate with each other and generate reliable droplet routes for the bioassays.

- We develop a MEDA simulator in PettingZoo Gym environment (Terry et al., 2021). We open-source the simulator to the RL community for future research[2].

2. Background

2.1. Multi-Agent Reinforcement Learning

MARL problems can be mathematically described using Markov/Stochastic Games (MGs) (Shapley, 1953). An MG is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \{\mathcal{O}^i\}_{i\in\mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, ..., N\}$ is the set of $N > 1$ agents; $\mathcal{S}$ is the state space; $\mathcal{A}^i$ denotes the action space of agent i; a probability function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \triangle(\mathcal{S})$ describes the transition probability from state $s_t \in \mathcal{S}$ to state $s_{t+1} \in \mathcal{S}$ given by a joint action $a \in \mathcal{A}$; $\mathcal{O}^i$ is the observation by agent i; $R^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function for agent i when it transitions from (s_t, a_t^i) to s_{t+1}; a variable $\gamma \in [0, 1]$ denotes the discount factor that trades off between the immediate and future rewards.

At time t, each agent $i \in \mathcal{N}$ executes an action a_t^i based on the observation o_t^i. The environment then transitions to s_{t+1} and rewards each agent i by $R^i(s_t, a_t, s_{t+1})$. The goal of agent i is to find the best policy π^i that will maximize the total reward received from the environment from a start state to an end state. The expected cumulative discounted reward is expressed as $U^i(t) = \mathbb{E}[\sum_t \gamma^t \cdot R^i(s_t, a_t, s_{t+1})]$. For continuous state and action spaces, this problem is intractable, but recent advances in deep RL employ deep neural networks to approximate the optimal policy (Silver et al., 2017; Schulman et al., 2017; Oroojlooy & Hajinezhad, 2019; Zhang et al., 2020).

2.2. RL Algorithms

We briefly describe three deep reinforcement learning algorithms that are used to evaluate our MARL framework; these algorithms are temporal-difference (TD), on-policy gradient descent, and off-policy actor-critic approaches.

2.2.1. Double Deep Q-Network

The deep Q-network (DQN) algorithm (Mnih et al., 2013) is a TD method that uses a neural network to approximate the state-action value function

$$Q(s, a) = \max_{\pi} \mathbb{E}[\sum_{0}^{\infty} \gamma^i r_{t+i} | s_t = s, a_t = a, \pi]$$

DQN relies on an experience replay dataset $\mathcal{D}_t = \{m_1, ..., m_t\}$, which stores the agent's experiences $m_t = (s_t; a_t; r_t; s_{t+1})$ to reduce correlations between observations. The experience consists of the current state s_t, the action the agent took a_t, the reward it received r_t, and the next state after transition s_{t+1}. The learning update at each iteration j uses a loss function based on the TD update:

$$L_j(\theta_j) = \mathbb{E}_{m_k \sim \mathcal{D}}[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta_j))^2]$$

[2] https://github.com/tcliang-tw/meda-env.git

where θ_j and θ^- are the parameters of the online Q-networks and the target network, respective, and the experiences m_k are sampled uniformly from $\mathcal{D}$. The parameters of the target network are fixed for a number of iterations while the online network $Q(s, a; \theta_j)$ is updated by gradient descent. In partially observable environments, an agent can only observe o_t instead of the entire state s_t. The experience replay is therefore updated as $m_t = (o_t; a_t; r_t; o_{t+1})$.

In DQN, the *max* operator uses the same values to select an action and evaluate an action, which can lead to overoptimistic value estimation (Hasselt, 2010). An improved method named *double DQN* was proposed to mitigate this problem (Van Hasselt et al., 2016). In double DQN, the loss function at iteration j is updated as:

$$L_j(\theta_j) = \mathbb{E}_{m_k \sim \mathcal{D}}[(r+\gamma Q(s', \text{argmax}_{a'} Q(s', a'; \theta_j); \theta^-) \\ - Q(s, a; \theta_j))^2]$$

2.2.2. Proximal Policy Optimization Algorithm

Proximal policy optimization (PPO) is an on-policy method that improves gradient-descent stability without performance collapse (Schulman et al., 2017). It updates policies using the following equation:

$$\theta_{k+1} = \underset{\theta}{\text{argmax}} \underset{s,a \sim \pi_{\theta_k}}{\mathbf{E}} [L(s, a, \theta_k, \theta)].$$

The update usually takes several steps of stochastic gradient descent (SGD) to maximize the objective. Here, the loss function L is defined as:

$$L(s, a, \theta_k, \theta) = \min(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a), g(\epsilon, A^{\pi_{\theta_k}}(s, a)))$$

where A is an estimator of the advantage function, ϵ is a hyperparameter, and

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & \text{if } A \geq 0 \\ (1-\epsilon)A & \text{if } A < 0 \end{cases}$$

2.2.3. Actor-Critic with Experience Replay

Actor-critic with experience replay (ACER) is an off-policy actor-critic model that increases the sample efficiency and reduces the data correlation (Wang et al., 2017). Similar to asynchronous advantage actor-critic (A3C) (Mnih et al., 2016), ACER learns the value function by training multiple actors in parallel. To obtain stability of the off-policy estimator, ACER adopts a retrace Q-value estimation:

$$\Delta Q^{ret}(S_t, A_t) = \gamma^t \prod_{1 \leq \gamma \leq t} \min(c, \frac{\pi(A_\tau|S_\tau)}{\beta(A_\tau|S_\tau)}) \delta_t$$

where (π, β) is the target and behavior policy pair, δ_t is the TD error, and c is a constant. In addition to a retrace Q-value estimation, ACER uses importance sampling and a trust region policy optimization (Schulman et al., 2015).

2.3. MARL Training Schemes

We consider three widely-used training schemes for our MARL framework: centralized, concurrent, and parameter-sharing (Gupta et al., 2017). We briefly describe how each approach can be used with MARL.

Centralized: The centralized learning approach assumes a joint model that receives all the observations and generates the joint actions for all the agents. A drawback of this approach is that it leads to an exponential growth in the observation and actions spaces with the number of agents.

Concurrent: In concurrent learning, each agent learns its own individual policy. Each independent policy maps an agent's private observation to an action. In the policy gradient approach, this means optimizing multiple policies simultaneously from the joint reward signal.

Parameter Sharing: Similar to concurrent learning, each agent is assigned with a neural network policy. However, in the parameter-sharing approach, all the agents share the parameters of a single policy. This allows the policy to be trained with the experiences of all agents simultaneously. However, each agent is still able to act differently based on the observation it receives.

2.4. Reward Structure

The concept of reward shaping (Ng et al., 1999) involves modifying rewards to accelerate learning without changing the optimal policy and approximate Bayesian methods (Kolter & Ng, 2009). In centralized learning, the reward cannot be decomposed into separate elements; this reward structure is equivalent to the joint reward in a decentralized partially observable Markov decision process (Spaan, 2012). However, in decentralized learning, an alternative local reward representation can be derived and assigned to each agent so that the reward assignment is in a more fine-grained manner. The work in (Bagnell & Ng, 2005) showed that such local information can help reduce the number of samples required for learning.

3. Parallel Droplet Routing in MEDA biochips using MARL

3.1. Formulation of Parallel Droplet Routing as MARL

We formulate the droplet-routing problem in MEDA biochips as an MARL framework where agents are fully cooperative. We consider a bioassay that is executed on a MEDA biochip, wherein the droplet locations are captured in real-time using the sensors integrated in the microelectrode cells (Li et al., 2017; Zhong et al., 2018). A controller, typically a desktop, laptop, or an FPGA board, is connected to the MEDA platform to shift the actuation bitstreams to

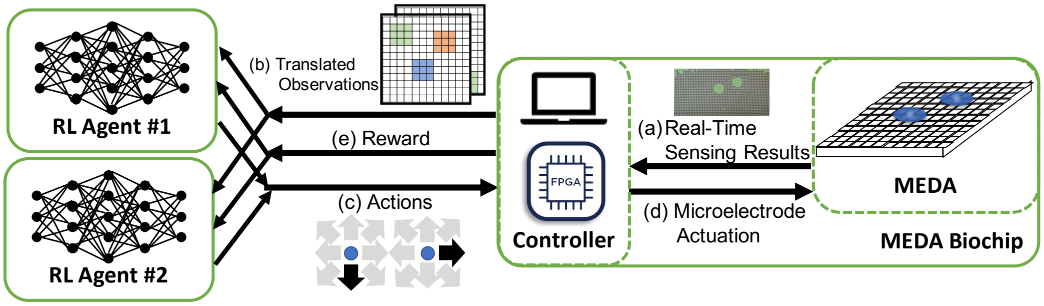

RL Agent #1
RL Agent #2
(b) Translated Observations
(e) Reward
(c) Actions
(a) Real-Time Sensing Results
(d) Microelectrode Actuation
FPGA
Controller
MEDA
MEDA Biochip

Figure 3. The MARL framework for parallel droplet routing on MEDA biochips. (a) Real-time sensing results are captured from MEDA. (b) Locations of the droplet are processed by the controller. The information is translated to arrays as the inputs for the RL agents. (c) The RL agents choose a set of actions. (d) The controller actuates microelectrodes based on the actions. (e) The RL agents receive a team-average reward.

MEDA and obtain the sensing results. The controller is also loaded with all the droplet-routing tasks to implement the actuation steps derived from bioasay synthesis (Lai et al., 2015). After obtaining a sensing result, the controller translates the environment states into observations for the agents. An agent can move a droplet to an adjacent location at any given time step, and the agent's goal is to transport the droplet from a given start location to a given destination. Based on the actions, the controller actuates on-chip droplets and observes the next state. The reward function for each agent is based on the state-transition result after an action is taken. Figure 3 provides an illustration of the overall MEDA system using the MARL framework, where two agents act on a MEDA environment.

Actions: At any time step, a droplet can be transported to any one of the eight directions: north, northeast, east, southeast, south, southwest, west, and northwest. The action set is defined as $A = \{a_n, a_{ne}, a_e, a_{se}, a_s, a_{sw}, a_w, a_{nw}\}$, where each element defines a direction that the droplet can be moved to.

Observations: At any given time step, the state of the MEDA-biochip is translated as RGB images, i.e., observation, for the agents. The resolution of the RGB image is the number of microelectrodes in the MEDA. The microelectrodes that are under the droplet to be routed are interpreted as blue pixels. The destination is defined as a set of microelectrodes, and they are interpreted as green pixels. During bioassay execution, multiple fluidic operations may be carried out concurrently to achieve high throughput. If a droplet is being transported while a concurrent mixing operation is also being carried out on the biochip, the electrodes that are used for the mixing operation cannot be used for droplet transportation to avoid undesirable contamination. The microelectrodes that are occupied by all the other concurrent operations and droplets are interpreted as red pixels. An example of RGB images is shown in Figure 3(b).

Rewards: We consider the cooperative setting for the MARL framework (Guériau et al., 2015; Li & Conitzer, 2015; Zhang et al., 2018; Doan et al., 2019) because the agents should not compete with each other to transport droplets. We first compute an assessment value r^i of an agent i after state transition. Let $\mathcal{R}^i$ be defined as radius (in terms of the number of microelectrodes) of droplet d^i and $e_{i,j}$ be the microelectrode at the i^{th} row and the j^{th} column of the MEDA. We assume that at time t, the center of droplet d^i is located at $e_{i,j}$, and its destination is at $e_{k,m}$. We define $D^i(t)$ as the distance of the droplet from the destination on the MEDA biochip at time t, where $D^i(t) = \sqrt{(i-k)^2 + (j-m)^2}$. After an action a^i_t is taken, if $D^i(t+1) < \mathcal{R}^i$, r^i is assigned a positive value of $+1.0$ because the droplet has reached the destination. Otherwise, the assessment value is computed as follows:

$$r^i = \begin{cases} -0.05 & \text{if } D^i(t+1) < D^i(t) \\ -0.1 & \text{if } D^i(t+1) \geq D^i(t) \end{cases}$$

In the first case, the action leads to a state in which the droplet is closer to the destination. In the second case, the action results in the same state or even a worse state. Therefore, we use a smaller value as the assessment value. In this reward setting, to gain the maximum value in a game, the agent is encouraged to take as few steps as possible to reach the destination.

As all the agents take a combination of actions, a possible resultant state is that droplets may get too close to each other, which can lead to unintended merging and sample/reagent contamination. To prevent this scenario, we also adjust the assessment values for droplets that are too close to each other. Assume that, after a joint set of actions is taken, the resultant locations of two droplets d^i and d^j are $e^{d^i}_{a,b}$ and $e^{d^j}_{c,d}$, respectively, and that the radii of the two droplets are $\mathcal{R}^i$ and $\mathcal{R}^j$, respectively. The distance of the two droplets is computed as $D(d^i, d^j) = \sqrt{(a-c)^2 + (b-d)^2}$. If $D(d^i, d^j) \leq 1.5 \times (\mathcal{R}^i + \mathcal{R}^j)$, the assessment values are adjusted as $r^i = r^i - 0.8$ and $r^j = r^j - 0.8$. In decentralized learning, each agent i is rewarded by its own assessment value r^i; in centralized learning, we give each agent a team-

Action	Result	Action	Result
a_N	$e^d_{i-\mathcal{R},j}$	a_S	$e^d_{i+\mathcal{R},j}$
a_{NE}	$e^d_{i-\mathcal{R}+1,j+\mathcal{R}-1}$	a_{SW}	$e^d_{i+\mathcal{R}-1,j-\mathcal{R}+1}$
a_E	$e^d_{i,j+\mathcal{R}}$	a_W	$e^d_{i,j-\mathcal{R}}$
a_{SE}	$e^d_{i+\mathcal{R}-1,j+\mathcal{R}-1}$	a_{NW}	$e^d_{i-\mathcal{R}+1,j-\mathcal{R}+1}$

Table 1. The transition outcome for a given action.

average reward $R_{avg} = \frac{\sum_{i=1}^{N} r^i}{N}$.

3.2. MEDA Simulator: Training of MARL Agents

To train the agents, we developed a PettingZoo-Gym environment named *MEDA-Env* to simulate a MEDA biochip. The MEDA matrix consists of $N \times M$ microelectrodes, where N and M are inputs to MEDA-Env. The actions, observations, and rewards of MEDA-Env are the same as the descriptions in Section 3.1. The transition model of MEDA-Env is described as below.

Transition model: Assuming that the center of a droplet d is located at $e_{i,j}$, we denote the location of the droplet as $e^d_{i,j}$. The radius of the droplet (in terms of the number of microelectrodes) is defined as $\mathcal{R}$ microelectrodes. The transition outcomes associated with all the actions are shown in Table 1. If the droplet is present at the boundary of the MEDA and the action is toward the outside of the biochip, the droplet will remain at the same location. For example, if the droplet for which the radius $\mathcal{R} = 2$ is present at $e_{2,2}$ and the action is either a_N or a_W, the droplet remains at $e_{2,2}$.

3.3. MARL Training

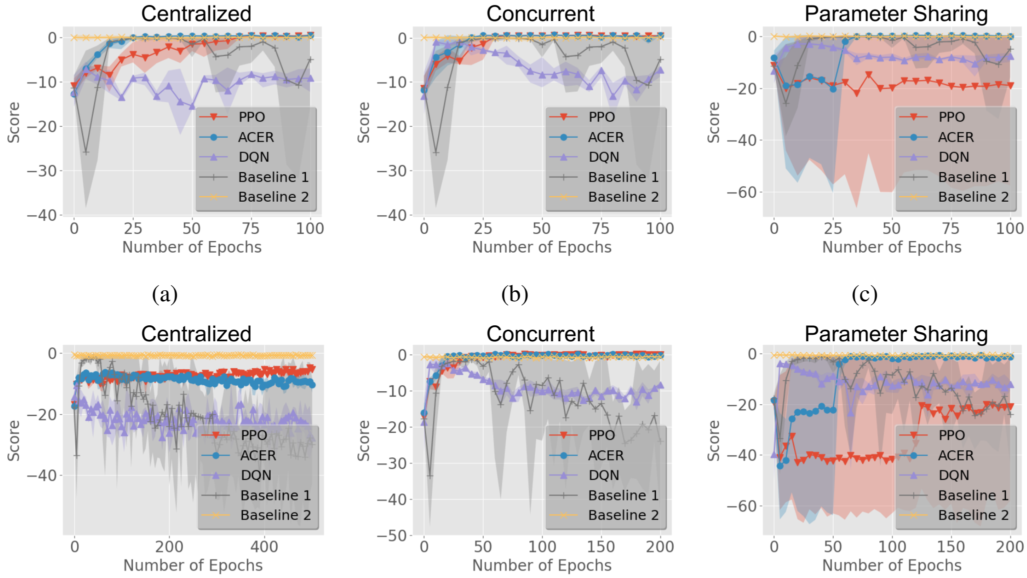
As shown in Figure 3, the RL agents are neural networks. Each agent i observes images as inputs and chooses an action $a^i_t \in A$. The agent receives a reward value based on the result of the previous joint actions. We used a simple but effective CNN for each agent because the network needs to be loaded on an affordable biochip platform. For example, in (Willsey et al., 2019), the cyberphysical biochip system includes only a quad-core 1.2 GHz ARMv7 processor with 1 GB RAM, and it does not contain a GPU; therefore, large networks are not feasible in this application scenario. A detailed description of the CNN can be found in the supplementary document.

We consider fabricated MEDA biochips as test cases and evaluate the effectiveness of RL-based adaptation using arrays of size 30 × 60 and 80 × 60 (Lai et al., 2015)[3]. We evaluated three RL algorithms, i.e. double DQN, PPO, and ACER, described in Section 2 using three training schemes, namely centralized, concurrent, and parameter sharing. The training was executed on a Linux platform integrated with a 11 GB-memory GPU (Nvidia GeForce RTX 2080 Ti). The training processes using PPO take ~ 2 hours to converge for decentralized learning, which is the fastest among the other algorithms. Although it takes several hours to train a model to perform as well as the offline method, the training process only needs to be carried out once, and the trained model can subsequently be used for all fabricated MEDA biochips. We compare the MARL approaches with two baseline methods: 1) the single-agent RL framework in (Liang et al., 2020a) and 2) a static (offline) routing method in (Keszocze et al., 2017).

We illustrate training processes for the MEDA-Env that contains 30×60 microelectrodes in Figure 4. For each RL algorithm, we ran 18 simulations with random seeds; the average performance of each algorithm is plotted as a solid line, and the similar color region shows the interval between its best performance and its worst performance. For each training game of MEDA-Env, n_{rt} random routing tasks are generated, where $1 < n_{rt} \leq 3$. A training epoch contains $20,000$ timesteps. We first see that single-agent RL (baseline 1) does not perform well because the agent does not know how to collaborate with other agents. We observe that double DQN does not converge in all training schemes. In some cases, double DQN learned sub-optimal policies first, and then the policy learned lower-reward experiences, which results in converging to more passive policies. The results are similar to MARL training in other environments (Jiang et al., 2020). We observe that PPO performs well in centralized and concurrent learning, but it sometimes cannot converge in parameter-sharing learning. This is because PPO is sensitive to initialization (Wang et al., 2019; Lazaridis et al., 2020; Hsu et al., 2020). In addition, the PPO update rule encourages the policy to exploit rewards that it has already found over the training course. Therefore, if an initial network policy is far from global optima, the policy can be easily trapped in local minima. We also observe that ACER is the most sample-efficient algorithm in decentralized learning. However, ACER encounters scalability issues in centralized learning, where reward becomes sparse in an exponentially growing action space. As the action space and observation space grow exponentially, the experiences stored in the limited replay buffer become important for ACER training. For centralized learning, our full training results show that PPO can outperform ACER after 200 training epochs; the training results are similar to that obtained in several Atari games (Fakoor et al., 2020).

We recorded a video of droplet routing by the MARL model (PPO in centralized learning) for a 30 × 60 MEDA during training, and it can be found in (Liang et al., 2021). From the video we see that, at first, the agents moved the droplets randomly without knowing the right policy needed to reach the destinations. In some games, the droplets got too close such that unintended merging and contamination happened. However, after 100K training games, the agents started to

[3] The supplementary document includes all the training processes.



(a) (b) (c)

(d) (e) (f)

Figure 4. Training process corresponding to different RL algorithms and training schemes. Score is the total reward that the MARL agents receive in a game. The performance is compared with two baseline methods: 1) a single-agent RL method (Liang et al., 2020a) and 2) a static (offline) routing method in (Keszocze et al., 2017). (a, b, c) Results with at most 2 concurrent routing tasks. (d, e, f) Results with at most 3 concurrent routing tasks.

"learn" from past experience. The agents started to keep safe distances between each other. After 160K training games, the agents could transport droplets to the corresponding destinations using the shortest path for some of the routing tasks. After 200K training games, the agents were able to complete all the routing tasks using the shortest paths.

4. MARL Evaluation When Microelectrode Degradation Occurs

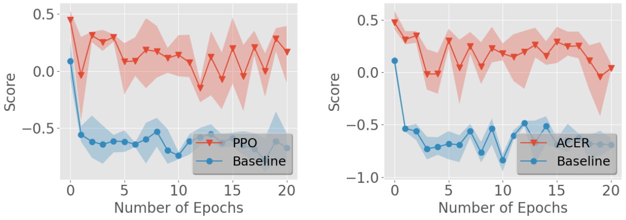
We evaluate the performance of the models in a realistic simulation setting, where microelectrodes degrade over time. We define a function $dg(e_{i,j})$ that describes the degradation status of a microelectrode, where $0 \leq dg(e_{i,j}) \leq 1$. If the microelectrode $e_{i,j}$ is healthy, $dg(e_{i,j}) = 1$; if the microelectrode $e_{i,j}$ has degraded, $dg(e_{i,j}) = 0$. The study in (Dong et al., 2015) showed that an electrode can only be actuated up to 200 times before it is completely degraded. Therefore, we define a degradation factor τ, where $0.6 \leq \tau < 1$, and the degradation function $dg(e_{i,j})$ is defined as $dg(e_{i,j}) = \tau^{\lfloor n/50 \rfloor}$, where n is the number of actuations. Each microelectrode is randomly assigned a different value of τ to simulate the variance of the microelectrode degradation in the microelectrode array.

A Bernoulli random variable X_{a_t} is defined as the transition outcome when droplet d takes an action a_t: when $X_{a_t} = 1$, the transition is successful as the normal transition function $T(e^d_{i,j}, a_t)$; when $X_{a_t} = 0$, the transition fails, and the droplet remains at the same location. Suppose a droplet d is present over m microelectrodes, and the set of these m microelectrodes is $S(d)$. The probability mass function of $X_{i,j}$ is defined as

$$\begin{cases} P(X_{i,j} = 1) = \dfrac{\sum_{e_{i,j} \in S(d)} dg(e_{i,j})}{m} \\ P(X_{i,j} = 0) = 1 - \dfrac{\sum_{e_{i,j} \in S(d)} dg(e_{i,j})}{m} \end{cases}$$

The training processes in Section 3.3 show that models in concurrent schemes are more sample-efficient than the other schemes. Therefore, we used the PPO and ACER models that have been trained to achieve the same performance as that of the baseline (Keszocze et al., 2017). Figure 5 shows the simulation results when $n_{rt} = 2$ and microelectrodes in MEDA-Env degrade over time[4]. We observe that the MARL model performs similar to the static (offline) method when the biochip starts to degrade. This is because the MARL model has been trained to perform as well as the baseline in the healthy mode of MEDA-Env. After the biochip is used for a while, we see that the performance of the baseline method degrades because the baseline method does not know which microelectrode is degraded and cannot dynamically change the routing paths. On the other hand, the MARL model "learns" the degradation process of the biochip and alter the routing paths accordingly. Therefore, the MARL model outperforms the baseline.

[4] Simulation results with three and four concurrent routing tasks are provided in the supplementary document.



(a) PPO (b) ACER

Figure 5. Comparison between the MARL agents and the baseline method in degrade mode with at most 2 routing tasks in a game.

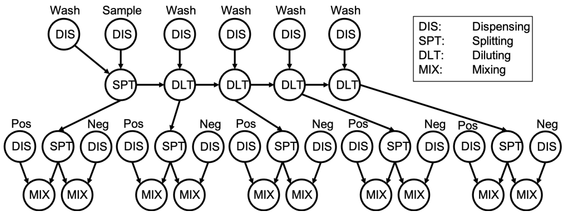


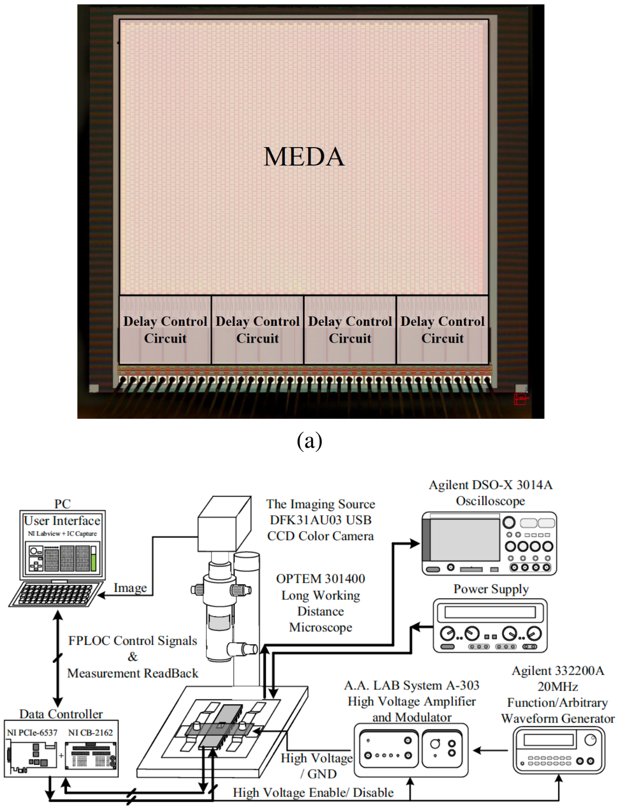
Figure 6. The steps involved in the serial-dilution bioassay.

5. Bioassay Execution on a Fabricated Biochip

The MARL framework can be used for any bioassay. We designed and executed a real-life bioassay, namely serial dilution, on a fabricated MEDA biochip because this benchtop bioassay requires large sample volumes and long execution time, and they are labor-intensive. Previous work has shown the effectiveness of related bioassays on MEDA biochips (Zhong et al., 2020). The executed bioassay contains 49 routing tasks, and we used the trained MARL droplet router to transport droplets.

5.1. Bioassay

Serial dilutions are widely used in experimental sciences, including biochemistry, pharmacology, microbiology, and physics (Ben-David & Davidson, 2014; Voller et al., 1976). A serial-dilution bioassay is used to accurately create solutions for experiments resulting in concentration curves with a logarithmic scale. The steps of a serial-dilution bioassay are shown in Figure 6. The sample is first mixed with the positive and negative reagents, and the reactions of the mixtures are recorded. Next, the sample is diluted with the buffer solution, and the diluted sample is mixed with the positive and negative reagents. This procedure is repeated until the sample is diluted to a desired concentration. The serial-dilution bioassay is often used for drug development such as in the case of antibiotics (Gullberg et al., 2011; Negreanu et al., 2012).


MEDA
Delay Control Circuit
Delay Control Circuit
Delay Control Circuit
Delay Control Circuit
PC
User Interface
Image
The Imaging Source DFK31AU03 USB CCD Color Camera
Agilent DSO-X 3014A Oscilloscope
OPTEM 301400 Long Working Distance Microscope
Power Supply
FPLOC Control Signals & Measurement ReadBack
A.A. LAB System A-303 High Voltage Amplifier and Modulator
Agilent 33220A 20MHz Function/Arbitrary Waveform Generator
Data Controller
NI PCIe-6537
NI CB-2162
High Voltage / GND
High Voltage Enable/ Disable

(a)

(b)

Figure 7. (a) The fabricated MEDA biochip. (b) The experimental setup.

5.2. Experimental Setup

We designed a 60 × 80 MEDA biochip for the experiment and fabricated the biochip using services at Taiwan Semiconductor Manufacturing Company (TSMC, 2021). The chip micro-photo and setup for the demonstration are shown in Figure 7. The chip has an area of $17.2\,\text{mm}^2$, and it was fabricated using a $0.35\,\mu\text{m}$ standard CMOS process. The chip was operated under 3.3 V at 1 KHz frequency. Reservoir modules are placed on the sides of the MEDA, and the modules can dispense different reagent droplets.

Figure 7(b) illustrates an overview of the experimental setup. The computer in the system is used to send actuation bitstreams, read the real-time sensing results, and run the pretrained MARL model. For droplet location sensing, the sensing results are extracted and processed using a digital signal processing scheme (Lai et al., 2015).

5.3. Experimental Results

We executed the droplet transportation of the bioassay on the fabricated MEDA biochip. Although there were a large number (49) of routing tasks that needed to be carried out on the MEDA, the MARL model was able to transport all the droplets to the destinations. During transportation, the agents kept safe distances each other so that unintended merging and contamination were avoided. We recorded the MARL-controlled routing operations in a video; it can be viewed in (Liang et al., 2021).

6. Conclusion

As digital microfluidic biochips are being adopted for point-of-care diagnostics such as COVID-19 testing, it is important to ensure that on-chip droplets are transported with a high degree of parallelism and reliably. We have presented a novel MARL framework for parallel droplet routing on MEDA biochips. To train an MARL model without using fabricated biochips, we have developed a PettingZoo-Gym environment that can be used to train the MARL droplet router for various MEDA sizes. The training process is a one-time effort, and the trained model can be used for many fabricated MEDA biochips. The experimental results showed that the MARL droplet router can reliably transport multiple droplets without unintended fluidic contamination. Our experimental results also showed that the trained MARL model can adapt to a dynamic environment, i.e., a degrading biochip, where microelectrodes degrade over time. In addition, we have demonstrated that using the MARL framework, a bio-protocol can run on a fabricated MEDA biochip.

Acknowledgement

This research was supported in part by the National Science Foundation under grants CCF-1702596 and ECCS-1914796, and the Ministry of Science and Technology (MOST), Taiwan, under grants 108-2221-E-009-071-MY3 and 109-2221-E-007-073-MY3.

References

Bagnell, D. and Ng, A. On local rewards and scaling distributed reinforcement learning. *Proceedings of the Advances in Neural Information Processing Systems*, 18: 91–98, 2005.

Ben-David, A. and Davidson, C. E. Estimation method for serial dilution experiments. *Journal of Microbiological Methods*, 107:214–221, 2014.

Brown, N. and Sandholm, T. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.

Chakrabarty, K., Fair, R. B., and Zeng, J. Design tools for digital microfluidic biochips: Toward functional diversification and more than moore. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(7):1001–1017, 2010.

Chou, W.-L., Lee, P.-Y., Yang, C.-L., Huang, W.-Y., and Lin, Y.-S. Recent advances in applications of droplet microfluidics. *Micromachines*, 6(9):1249–1271, 2015.

Doan, T. T., Maguluri, S. T., and Romberg, J. Finite-time analysis of distributed TD(0) with linear function approximation for multi-agent reinforcement learning. *arXiv preprint arXiv:1902.07393*, 2019.

Dong, C., Chen, T., Gao, J., Jia, Y., Mak, P.-I., Vai, M.-I., and Martins, R. P. On the droplet velocity and electrode lifetime of digital microfluidics: Voltage actuation techniques and comparison. *Microfluidics and Nanofluidics*, 18(4):673–683, 2015.

Drygiannakis, A. I., Papathanasiou, A. G., and Boudouvis, A. G. On the connection between dielectric breakdown strength, trapping of charge, and contact angle saturation in electrowetting. *Langmuir*, 25(1):147–152, 2009.

Fakoor, R., Chaudhari, P., and Smola, A. J. P3O: Policy-on policy-off policy optimization. In *Proceedings of the Uncertainty in Artificial Intelligence*, pp. 1017–1027. PMLR, 2020.

Ganguli, A., Mostafa, A., Berger, J., Aydin, M. Y., Sun, F., Ramirez, S. A. S. d., Valera, E., Cunningham, B. T., King, W. P., and Bashir, R. Rapid isothermal amplification and portable detection system for SARS-CoV-2. *Proceedings of the National Academy of Sciences*, 2020.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *Proceedings of the International Conference on Robotics and Automation*, pp. 3389–3396. IEEE, 2017.

Guériau, M., Billot, R., El Faouzi, N.-E., Hassas, S., and Armetta, F. Multi-agent dynamic coupling for cooperative vehicles modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Gullberg, E., Cao, S., Berg, O. G., Ilbäck, C., Sandegren, L., Hughes, D., and Andersson, D. I. Selection of resistant bacteria at very low antibiotic concentrations. *PLoS Pathog*, 7(7):e1002158, 2011.

Gupta, J. K., Egorov, M., and Kochenderfer, M. Cooperative multi-agent control using deep reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 66–83. Springer, 2017.

Hasselt, H. Double Q-learning. *Proceedings of the Advances in Neural Information Processing Systems*, 23: 2613–2621, 2010.

Ho, Y., Wang, G., Lai, K. Y.-T., Lu, Y.-W., Liu, K.-M., Wang, Y.-M., and Lee, C.-Y. Design of a micro-electrode cell for programmable lab-on-CMOS platform. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2871–2874, 2016.

Hsu, C. C.-Y., Mendler-Dünner, C., and Hardt, M. Revisiting design choices in proximal policy optimization. *arXiv preprint arXiv:2009.10897*, 2020.

Inc., B. Baebies Official Website. https://baebies.com, 2021. [Online; accessed 30-January-2021].

Jiang, J., Dun, C., Huang, T., and Lu, Z. Graph convolutional reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2020.

Keszocze, O., Li, Z., Grimmer, A., Wille, R., Chakrabarty, K., and Drechsler, R. Exact routing for micro-electrode-dot-array digital microfluidic biochips. In *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 708–713. IEEE, 2017.

Kolter, J. Z. and Ng, A. Y. Near-Bayesian exploration in polynomial time. In *Proceedings of the International Conference on Machine Learning*, pp. 513–520, 2009.

Lai, K. Y.-T. et al. An intelligent digital microfluidic processor for biomedical detection. *Journal of Signal Processing Systems*, 78(1):85–93, 2015.

Lazaridis, A., Fachantidis, A., and Vlahavas, I. Deep reinforcement learning: A state-of-the-art walkthrough. *Journal of Artificial Intelligence Research*, 69:1421–1471, 2020.

Lee, J., Kim, R., Yi, S. W., and Kang, J. Maps: Multi-agent reinforcement learning-based portfolio management system. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 4520–4526. International Joint Conferences on Artificial Intelligence, 2020.

Li, J. et al. Current commercialization status of electrowetting-on-dielectric (EWOD) digital microfluidics. *Lab on a Chip*, 20(10):1705–1712, 2020.

Li, Y. and Conitzer, V. Cooperative game solution concepts that maximize stability under noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 979–985, 2015.

Li, Z., Lai, K. Y.-T., Yu, P.-H., Chakrabarty, K., Ho, T.-Y., and Lee, C.-Y. Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips. *IEEE Transactions on Biomedical Circuits and Systems*, 11(3):612–626, 2017.

Liang, T.-C., Zhong, Z., Bigdeli, Y., Ho, T.-Y., Chakrabarty, K., and Fair, R. Adaptive droplet routing in digital microfluidic biochips using deep reinforcement learning. In *Proceedings of International Conference on Machine Learning*, 2020a.

Liang, T.-C., Zhong, Z., Pajic, M., and Chakrabarty, K. Extending the lifetime of MEDA biochips by selective sensing on microelectrodes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3531–3543, 2020b.

Liang, T.-C., Zhou, J., Chan, Y.-S., Ho, T.-Y., Chakrabarty, K., and Lee, C.-Y. Recorded Videos during Training and Evaluation. https://drive.google.com/drive/folders/1yVhI7-T5YaC2D5a_d78sMuFmFhgl_HZq?usp=sharing, 2021. [Online; accessed 23-May-2021].

Manak, M. S., Varsanik, J. S., Hogan, B. J., Whitfield, M. J., Su, W. R., Joshi, N., Steinke, N., Min, A., Berger, D., Saphirstein, R. J., et al. Live-cell phenotypic-biomarker microfluidic assay for the risk stratification of cancer patients via machine learning. *Nature Biomedical Engineering*, 2(10):761–772, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 1928–1937. PMLR, 2016.

Negreanu, Y., Pasternak, Z., Jurkevitch, E., and Cytryn, E. Impact of treated wastewater irrigation on antibiotic resistance in agricultural soils. *Environmental Science & Technology*, 46(9):4800–4808, 2012.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning*, volume 99, pp. 278–287, 1999.

NIH. Rapid acceleration of diagnostics. https://www.nih.gov/research-training/medical-research-initiatives/radx, 2021a. [Online; accessed 30-January-2021].

NIH. Nih delivering new covid-19 testing technologies to meet u.s. demand. https://www.nih.gov/news-events/news-releases/nih-delivering-new-covid-19-testing-technologies-meet-us-demand, 2021b. [Online; accessed 30-January-2021].

Oroojlooy, A. and Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963*, 2019.

Ouyang, Y., Yin, J., Wang, W., Shi, H., Shi, Y., Xu, B., Qiao, L., Feng, Y., Pang, L., Wei, F., et al. Down-regulated gene expression spectrum and immune responses changed during the disease progression in covid-19 patients. *Clinical Infectious Diseases*, 2020.

Palanisamy, P. Multi-agent connected autonomous driving using deep reinforcement learning. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–7, 2020. doi: 10.1109/IJCNN48605.2020.9207663.

Pollack, M. G., Fair, R. B., and Shenderov, A. D. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Applied Physics Letters*, 77(11): 1725–1726, 2000.

Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017.

Schmitz, J. E. and Tang, Y.-W. The GenMark ePlex®: another weapon in the syndromic arsenal for infection diagnosis. *Future microbiology*, 13(16):1697–1708, 2018.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning*, pp. 1889–1897, 2015.

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

Shapley, L. S. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

Sheridan, C. Covid-19 spurs wave of innovative diagnostics. *Nature biotechnology*, 38(7):769–772, 2020.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of GO without human knowledge. *Nature*, 550(7676):354–359, 2017.

Sista, R. S. et al. Digital microfluidic platform to maximize diagnostic tests with low sample volumes from newborns and pediatric patients. *Diagnostics*, 10(1):21, 2020.

Spaan, M. T. Partially observable markov decision processes. In *Reinforcement Learning*, pp. 387–414. Springer, 2012.

Su, F., Ozev, S., and Chakrabarty, K. Test planning and test resource optimization for droplet-based microfluidic systems. *Journal of Electronic Testing*, 22(2):199–210, 2006.

Sun, F., Ganguli, A., Nguyen, J., Brisbin, R., Shanmugam, K., Hirschberg, D. L., Wheeler, M. B., Bashir, R., Nash, D. M., and Cunningham, B. T. Smartphone-based multiplex 30-minute nucleic acid test of live virus from nasal swab extract. *Lab on a Chip*, 20(9):1621–1627, 2020.

Terry, J. K., Black, B., Jayakumar, M., Hari, A., Santos, L., Dieffendahl, C., Williams, N. L., Lokesh, Y., Sullivan, R., Horsch, C., and Ravi, P. Pettingzoo: Multi-agent reinforcement learning environments. https://www.pettingzoo.ml, 2021. [Online; accessed 30-January-2021].

TSMC. Taiwan Semiconductor Manufacturing Company. https://www.tsmc.com/english/default.htm, 2021. [Online; accessed 30-January-2021].

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Verheijen, H. and Prins, M. Reversible electrowetting and trapping of charge: Model and experiments. *Langmuir*, 15(20):6616–6620, 1999.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Voller, A., Bidwell, D., and Bartlett, A. Enzyme immunoassays in diagnostic medicine: Theory and practice. *Bulletin of the World Health Organization*, 53(1):55, 1976.

Wachi, A. Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving. In *Proceedings of the International Joint Conference on on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 2019.

Wang, G., Teng, D., and Fan, S.-K. Digital microfluidic operations on micro-electrode dot array architecture. *IET Nanobiotechnology*, 5(4):152–160, 2011.

Wang, Y., He, H., Tan, X., and Gan, Y. Trust region-guided proximal policy optimization. In *Proceedings of the Advances in Neural Information Processing Systems*, 2019.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. Sample efficient actor-critic with experience replay. In *Proceedings of the International Conference on Learning Representations*, 2017.

Weng, L. Policy gradient algorithms. https://lilianweng.github.io/lil-log/2018/04/08/policy-gradient-algorithms.html, 2018. [Online; accessed 30-January-2021].

Willsey, M., Stephenson, A. P., Takahashi, C., Vaid, P., Nguyen, B. H., Piszczek, M., Betts, C., Newman, S., Joshi, S., Strauss, K., et al. Puddle: A dynamic, error-correcting, full-stack microfluidics platform. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 183–197, 2019.

Xu, T. and Chakrabarty, K. Functional testing of digital microfluidic biochips. In *Proceedings of the International Test Conference*, pp. 1–10. IEEE, 2007a.

Xu, T. and Chakrabarty, K. Integrated droplet routing in the synthesis of microfluidic biochips. In *Proceedings of the Annual Design Automation Conference*, pp. 948–953, 2007b.

Zhang, H., Chen, W., Huang, Z., Li, M., Yang, Y., Zhang, W., and Wang, J. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Başar, T. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757*, 2018.

Zhang, K., Yang, Z., and Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Zhong, Z., Li, Z., Chakrabarty, K., Ho, T.-Y., and Lee, C.-Y. Micro-electrode-dot-array digital microfluidic biochips: Technology, design automation, and test techniques. *IEEE Transactions on Biomedical Circuits and Systems*, 13(2):292–313, 2018.

Zhong, Z., Liang, T.-C., and Chakrabarty, K. Enhancing the reliability of MEDA biochips using IJTAG and wear leveling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.