Adaptive Droplet Routing in Digital Microfluidic Biochips Using Deep Reinforcement Learning

Tung-Che Liang [1] Zhanwei Zhong [1] Yaas Bigdeli [1] Tsung-Yi Ho [2] Krishnendu Chakrabarty [1] Richard Fair [1]

Abstract

We present and investigate a novel application domain for deep reinforcement learning (RL): droplet routing on digital microfluidic biochips (DMFBs). A DMFB, composed of a two-dimensional electrode array, manipulates discrete fluid droplets to automatically execute biochemical protocols such as point-of-care clinical diagnosis. However, a major concern associated with the use of DMFBs is that electrodes in a biochip can degrade over time. Droplet-transportation operations associated with the degraded electrodes can fail, thereby compromising the integrity of the bioassay outcome. We show that casting droplet transportation as an RL problem enables the training of deep network policies to capture the underlying health conditions of electrodes and provide reliable fluidic operations. We propose a new RL-based droplet-routing flow that can be used for various sizes of DMFBs, and demonstrate reliable execution of an epigenetic bioassay with the RL droplet router on a fabricated DMFB. To facilitate further research, we also present a simulation environment based on the OpenAI Gym Interface for RL-guided droplet-routing problems on DMFBs.

1. Introduction

There has been rapid progress in recent years on using deep neural networks trained by reinforcement learning (RL). These systems have outperformed humans in games (Mnih et al., 2013; Silver et al., 2017), and they have also shown tremendous promise in robotics (Gu et al., 2017) and natural language processing (He et al., 2016; Narasimhan et al., 2015). This is because RL-guided systems that are cast in dynamic environments can learn from previous experience and improve their reaction to the environment. In this paper, we show that since the electrode condition of a digital microfluidic biochip (DMFB) dynamically changes over time, we can exploit advances in RL to ensure more reliable droplet transportation in DMFBs.

1.1. Digital Microfluidic Biochips

A digital microfluidic biochip (DMFB) is composed of a two-dimensional electrode array that manipulates discrete fluid droplets. When driven by a sequence of control voltages, the electrode array can perform fluidic operations, such as dispensing, mixing, and splitting (Choi et al., 2012; Ho et al., 2010). Because of the precise control over microfluidic operations, DMFBs are employed in lab-on-a-chip systems for high-throughput DNA sequencing and point-of-care clinical diagnosis (Myers & Lee, 2008; Liu et al., 2013; Lehotay & Cook, 2015; Perut et al., 2016). Figure 1(a) shows a DMFB, where two droplets are present on a patterned electrode array.

DMFBs manipulate nanoliter droplets using the principle of *electrowetting-on-dielectric* (EWOD) (Pollack et al., 2000). EWOD refers to the modulation of the interfacial tension between a conductive fluid and a solid electrode coated with a dielectric layer by applying an electric field between them. As shown in Figure 1(b), an imbalance of interfacial tension is created if an electric field is applied to only one side of the droplet; this interfacial tension gradient forces the droplet to move toward to the right.

Illumina, a market leader in DNA sequencing, transitioned digital microfluidics to the marketplace for sample preparation in 2015 through NeoPrep—a $\sim \$40K$ instrument to automate the preparation of up to 16 sequencing libraries at a time (Illumina, 2015). This technology has also been deployed by Genmark for infectious disease testing (Pierce & Hodinka, 2012) and by Baebies to detect lysosomal storage diseases in newborns (Hopkins et al., 2015). These commercialization success stories highlight the emergence of DMFB technology in the marketplace.

However, reliability is still a critical concern in DMFB systems. Illumina announced in February 2017 that it is immediately halting sales of NeoPrep. In its letter to customers, Illumina cited reliability issues in-house and far worse ones in the field. While biochips are tested immediately after production to ensure biochip integrity, biochip defects, such as electrode degradation, can occur throughout the lifetime

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

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

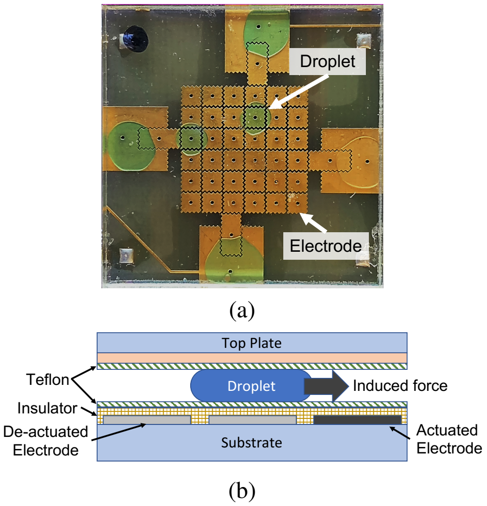


Figure 1. (a) Top-view of a DMFB. Two droplets are present on the biochip. (b) Illustration of the side-view of a DMFB. The droplet is moved to the right using EWOD.

of the system (Su & Chakrabarty, 2006; Dong et al., 2015). Electrode degradation results from charge trapping in the dielectric insulator (Chen et al., 2013). A consequence of electrode degradation is that droplets' motion is prevented because of the unwanted variation of surface-tension forces along their flow path (Su et al., 2006a). An example of electrode degradation is shown in Figure 2 for a fabricated biochip. Two droplets are present on the biochip, and one of them is present over a degraded electrode. In the next time slot, two electrodes are actuated to move the two droplets. However, one of the two fluidic operations fails because unwanted surface-tension force is extended by the degraded electrode. Detailed analyses of the correlation between electrode defects and fluidic operations have been presented in (Drygiannakis et al., 2008).

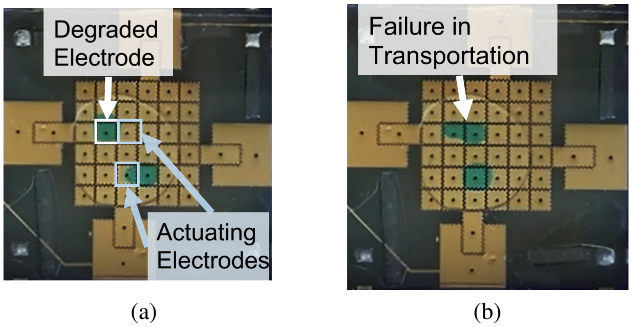


Figure 2. Droplet transportation fails because of electrode degradation. (a) Two droplets are present on the electrode array. Two electrodes are actuated to move the two droplets. (b) After electrode actuation, the upper droplet cannot be moved to the desired electrode completely because it was present over a degraded electrode; the lower droplet is transported to the desired electrode correctly.

1.2. Motivating RL-Guided Droplet Routing

In the DMFB synthesis flow (Su & Chakrabarty, 2004), a bioassay protocol with specified fluidic operations is first developed by biologists. Then, a synthesis tool maps fluidic operations to areas of electrodes, referred to as fluidic modules, of a biochip to perform the operations (Chakrabarty et al., 2010). 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 determining paths of droplet transportation between modules is referred to as *droplet routing*. Many droplet routing methods have been proposed to accomplish routing tasks in bioassay applications (Su et al., 2006b; Xu & Chakrabarty, 2007; Zhao & Chakrabarty, 2012). However, these methods are static and they neglect the fact that droplet transportation may fail if the electrodes associated with the routing path degrade over time.

Example: Consider a routing path that has been determined for bioassay execution as shown in Figure 3(a). This route is the shortest path between the start and the destination. In this case, the droplet can be transported to the destination because the biochip is healthy, i.e., no electrode degradation occured. Conversely, as shown in Figure 3(b), droplet transportation to the destination fails because some degraded electrodes are involved in the path. The droplet is likely to be stuck in this nonfunctional area. Assuming that an online droplet router knows where the degraded electrodes are, it can generate another route that involves only healthy electrodes. Figure 3(c) shows this alternative route, which is also the shortest path and avoids any degraded electrode.

In the above example, the degraded electrodes are marked with a different color. However, in reality, we cannot identify degraded electrodes by examining their appearance since the degradation process results from charge trapped in the insulator. Therefore, prior work was focused on synthesis methods that prevent excessive usage over a few electrodes by evenly distributing the fluidic operations to every electrode (Chen et al., 2013; Zhong et al., 2020). However, no method has been presented yet to overcome routing failure associated with electrode degradation. If electrodes degradation happens in a biochip during bioassay execution and one of the routing tasks is associated with the degraded electrodes, the bioassay execution will fail, and the bioassay will need to be re-executed on a new biochip (Huang et al., 2011). Furthermore, the locations of the degraded electrodes may vary from one biochip to another because the electrode-degradation process is affected by geometric variance of the electrode array and the differences in electrode actuation times (Ho et al., 2011).

An RL-based droplet router can, however, overcome the electrode-degradation problem and provide more reliable bioassay executions in three ways. First, the RL-based

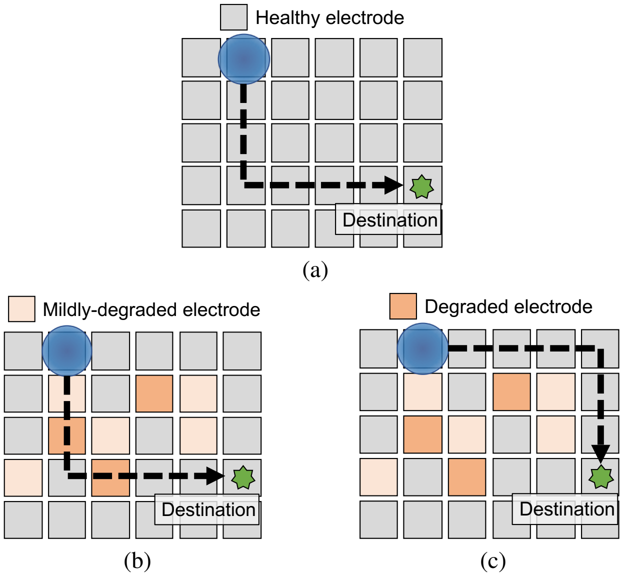


Figure 3. Droplet-routing paths from a start point to an end point. Healthy electrodes are colored gray; degraded electrodes are colored brown. (a) A pre-determined path for a healthy DMFB. (b) A DMFB has been used for a while, and some electrodes have degraded. Some degraded electrodes are involved in the pre-determined path. Droplet transportation may fail. (c) A more reliable path for the aged DMFB.

droplet router provides real-time decisions for droplet routes. Second, the RL-based droplet router can "learn" from the past experience associated with electrodes that start malfunctioning. Therefore, the droplet router can generate routing paths that involve healthy electrodes. Third, the online RL-based droplet router can adapt to any degradation conditions. Even though degradation processes on one DMFB may be different from that of another, the droplet router can generate different, yet reliable, routing paths on two distinct DMFBs given the same routing task.

1.3. Paper Contributions

Our research is the first attempt to apply RL to emerging microfluidic systems. The key contributions of this paper are as follows.

- We formulate a novel framework for RL-based droplet routing on DMFBs. We discuss the challenges involved with casting droplet routing as an RL task.
- We present an online droplet routing framework that employs deep RL to generate a policy, which maps real-time observations of a DMFB, to dynamically choose droplet paths. In our framework, the policy is first trained in a simulated DMFB. The pre-trained policy is then loaded on the controller associated with a DMFB, and the policy generates real-time droplet routing paths.
- We demonstrate the routing scheme by executing an epigenetic bio-protocol on a fabricated DMFB. We show that the policy can learn the underlying degradation of electrodes and generate reliable routes for the bioassays.
- We develop a DMFB simulator in OpenAI Gym environment. We open-source the simulator to the RL community for future research[1].

2. RL Approach to Droplet Router on DMFBs

In this section, we provide an overview of RL and formulate the droplet routing on DMFBs as an RL problem.

2.1. Background

In RL, an agent is situated in an environment. The agent's goal is to accomplish a given task with the best performance given a set of small actions. At each step, the agent takes one of the given actions, and the agent receives an observation and reward from the environment (Sutton & Barto, 2018).

RL problems can be mathematically described using Markov decision processes (MDPs). An MDP contains two sets, namely S and A, a probability function f, a reward model R, and a variable γ. The observations made by the agent are included in a set S, and an observation is also referred to as a state. An element $s_t \in S$ is an observation made by the agent at time t toward the environment. We use A to denote a set of actions that are made by the agent. An action $a_t \in A$ denotes the action made by the agent at time t. We use $P(s_{t+1}|a_t, s_t)$ as the transition model, which describes what the next state s_{t+1} will be after the agent takes action a_t while in the current state s_t. The reward model is denoted by $R(s_t)$; it describes the reward that the agent receives when it enters the state s_t. The parameter γ is a discount factor, where $0 \leq \gamma \leq 1$ and $\gamma \in \mathbb{R}$. This factor represents the relative importance between immediate and future rewards. The agent's goal is to pick the best policy π that will maximize the total reward received from the environment from the start state to an end state. The expected cumulative discounted reward is expressed as $U(t) = \mathbb{E}[\sum_t \gamma^t \cdot R(s_t)]$. For large or 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).

2.2. Formulation of Droplet Routing as RL

We formulate the droplet-routing problem in DMFBs as a sequence of decision-making problems within the RL framework. We consider a droplet-routing agent that is able to make real-time observations of the DMFB. The agent can move the droplet to an adjacent electrode at a time step, and the agent's goal is to transport the droplet from a given start electrode to a given destination electrode. We reward or punish the agent based on the state-transition result after an

[1] https://github.com/tcliang-tw/dmfb-env.git

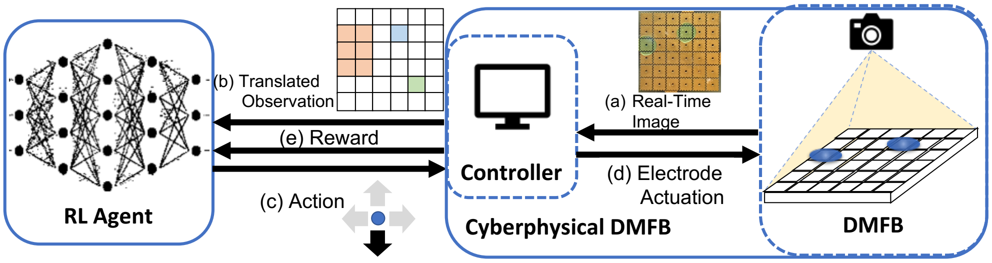


Figure 4. The RL framework for droplet routing on DMFBs. (a) Real-time images of the DMFB are captured using the CCD camera. (b) Locations of the droplet are processed by the controller. The information is translated to an array as the input for the RL agent. (c) The RL agent chooses an action. (d) The controller actuates electrodes based on the action. (e) The RL agent receives a reward accordingly.

action is taken.

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

States: A state s_t consists of the location of the transported droplet, the destination of the droplet, and electrodes that are occupied by other concurrent fluidic operations. 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 a DMFB, the electrodes that are used for the mixing operation cannot be used for droplet transportation to avoid undesirable contamination.

Rewards: The agent is rewarded if the droplet is transported to the destination. Let $e_{i,j}$ be the i^{th} row and the j^{th} column electrode of the DMFB. We assume that at state s_t, a droplet is present at $e_{i,j}$, and its destination is at $e_{k,m}$. We define $D(s_t)$ as the Manhattan distance of the droplet from the destination on the electrode array at state s_t, where $D(s_t) = |i-k| + |j-m|$. After an action a_t is taken, the agent is rewarded if $D(s_{t+1}) < D(s_t)$. On the contrary, the agent is punished if $D(s_{t+1}) \geq D(s_t)$ after a_t is taken.

3. Online Droplet Router

We consider a bioassay that is executed on a cyberphysical DMFB, wherein the droplet location is captured in real-time with a CCD camera (Luo et al., 2012; Willsey et al., 2019). A controller is connected to the DMFB and loaded with all the droplet-routing tasks to achieve the bioassay-synthesis result (Su et al., 2006a). Figure 4 provides an illustration of the overall system.

We propose an online droplet router, as the agent in the RL framework, that can accomplish all the droplet-routing tasks. To train the agent, we developed an OpenAI-Gym environment named *DMFB-Env* to simulate a cyberphysical DMFB. The DMFB matrix consists of $N \times M$ electrodes, where N and M are inputs to DMFB-Env.

3.1. DMFB Environment

States: At any given time step, the DMFB observation is processed as a simple RGB image. The locations of on-chip droplets are identified using control software (Luo et al., 2012). The resolution of the RGB image is the number of electrodes on the DMFB. A droplet-containing electrode is interpreted as a blue pixel. The destination electrode is interpreted as a green pixel. The electrodes that are occupied by all the other concurrent operations are interpreted as red pixels. An example of an RGB image is shown in Figure 4(b).

Transition model: DMFB-Env can be simulated in two modes: (1) healthy and (2) degrading. We first consider the healthy mode. Let $e_{i,j}$ be defined as earlier, an electrode at the i^{th} row and the j^{th} column of the DMFB. We denote the location of the droplet using $e_{i,j}$. The transition function is defined as

$$T(e_{i,j}, a_t) = \begin{cases} e_{i-1,j} & \text{if } a_t = a_N \\ e_{i+1,j} & \text{if } a_t = a_S \\ e_{i,j+1} & \text{if } a_t = a_E \\ e_{i,j-1} & \text{if } a_t = a_W \end{cases}$$

where $1 < i < N$ and $1 < j < M$. If the droplet is present at the boundary of the electrode array and the action is toward the outside of the biochip, the droplet will remain at the same location. For example, if the droplet is present at $e_{0,0}$ and the action is either a_N or a_W, the droplet remains at $e_{0,0}$. Similarly, if the next location of the droplet is in the electrodes that are used for the other concurrent fluidic operations, the droplet stays at the same electrode.

In the degrading mode, we define a function $d(e_{i,j})$ that describes the degradation status of an electrode, where $0 \leq d(e_{i,j}) \leq 1$. If the electrode $e_{i,j}$ is healthy, $d(e_{i,j}) = 1$; if the electrode $e_{i,j}$ has degraded, $d(e_{i,j}) = 0$. The study in (Dong et al., 2015) showed that in some extreme cases, 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 $d(e_{i,j})$ is defined as

$$d(e_{i,j}) = \tau^{\lfloor n/50 \rfloor}$$

where n is the number of actuations. Each electrode is randomly assigned a different value of τ to simulate the geometric variance of the electrode array.

A Bernoulli random variable $X_{i,j}$ is defined as the transition outcome when the droplet is present at $e_{i,j}$: when $X_{i,j} = 1$, the transition is successful as $T(e_{i,j}, a_t)$; when $X_{i,j} = 0$, the transition fails, and the droplet remains at the same electrode. The probability mass function of $X_{i,j}$ is defined as

$$\begin{cases} P(X_{i,j} = 1) = d(e_{i,j}) \\ P(X_{i,j} = 0) = 1 - d(e_{i,j}) \end{cases}$$

Reward function: Let $D(s_t)$ be as defined earlier, the Manhattan distance from the droplet to the destination on the electrode array at state s_t. After an action a_t is taken, if $D(s_{t+1}) = 0$, the agent receives a positive reward of $+1.0$. Otherwise, the reward is computed as follows

$$R_t = \begin{cases} +0.5 & \text{if } D(s_{t+1}) < D(s_t) \\ -0.3 & \text{if } D(s_{t+1}) = D(s_t) \\ -0.8 & \text{if } D(s_{t+1}) > D(s_t) \end{cases}$$

In the first case, the action leads to a state in which the droplet is closer to the destination. Therefore, we reward the agent with a positive value. Note that any positive value used here can lead to agent convergence because the agent will always maximize the total reward. In the second case, the agent is punished with a negative value because the action does not result in a better state. In the third case, the agent is punished with a value of -0.8 because it leads to a worse state. This negative value needs to be less than the negative value of the reward for the first case. Assuming that the negative value is -0.3, the agent can earn a positive reward by repeatedly leaving and re-entering a state ($-0.3 + 0.5 = 0.2$), i.e., this undesirable behavior is encouraged.

3.2. RL Agent

As shown in Figure 4, the RL agent is a deep neural network. It observes images as input and chooses an action $a_t \in A$. The agent receives a reward value based on the result of the previous action.

Neural network: Over the past few years, many neural network architectures have been proposed (LeCun et al., 1998; Simonyan & Zisserman, 2014; Howard et al., 2017). Because DMFBs commercially available today typically include a few hundred electrodes (Zhao & Chakrabarty, 2012), we evaluate the effectiveness of RL-based adaptation using DMFBs of size $N \times M$, where $25 \leq N \times M \leq 1,225$. We first evaluated several fully-connected neural networks. We found out that while fully-connected neural networks are effective for the droplet-routing problem for small DMFB instances (less than 100 electrodes), they do not converge for large DMFBs. We found that convolutional neural networks (CNNs) are effective for all the DMFB instances that we considered. However, because the network needs to be loaded on a cyberphysical DMFB, the computational resources on the associated controller may be limited compared to a server. For example, in (Willsey et al., 2019), the cyberphysical DMFB includes only a quad-core 1.2 GHz ARMv7 processor with 1 GB RAM, and it does not contain a GPU, therefore large CNNs are not feasible in this application scenario. We tested several options for the number of hidden layers and number of neurons per layer. We found that a simple CNN, as described in Table 1, can solve the droplet-routing problem for large DMFBs with more than $1,000$ electrodes.

Table 1. The convolutional neural network configuration.

Layer	Type	depth	Activation	Stride	Padding
1	Convolution	32	ReLU	3	1
2	Convolution	64	ReLU	3	1
3	Convolution	64	ReLU	3	1
4	Fully-Connected	256	ReLU	N/A	N/A
5	Fully-Connected	4	ReLU	N/A	N/A

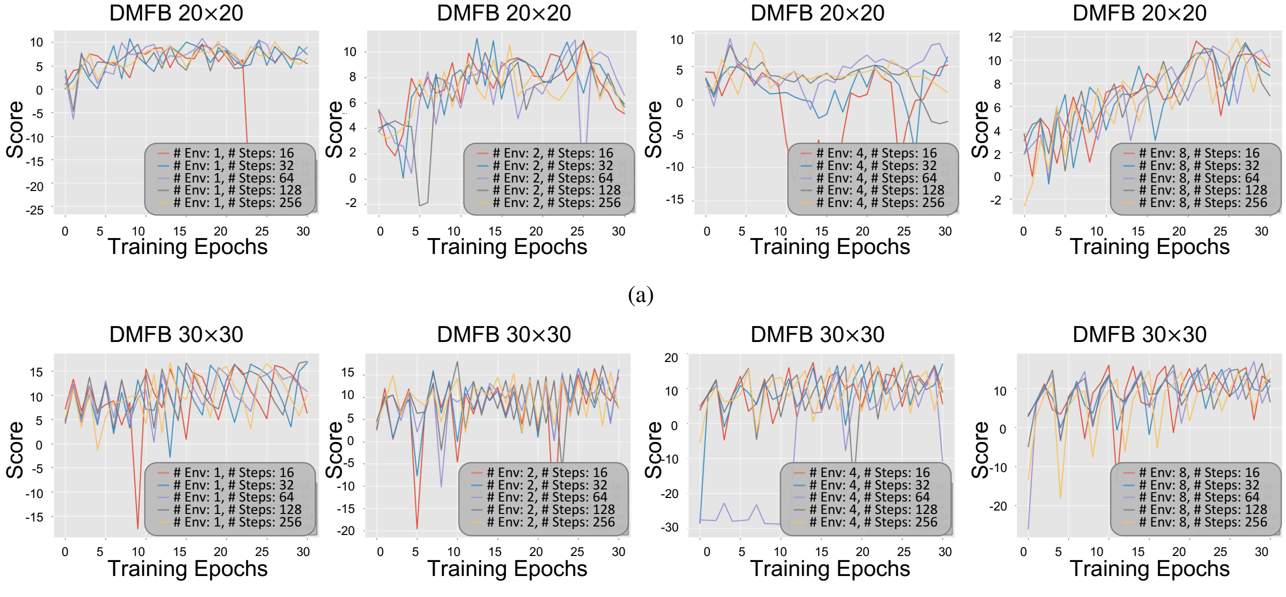
3.3. RL Training

We first trained our model in DMFB-Env in the healthy mode using the proximal policy optimization (PPO) algorithm (Schulman et al., 2017). The PPO algorithm combines the idea of having multiple workers to stabilize the training process (Mnih et al., 2016) and the idea of using a trust region to improve the actor (Schulman et al., 2015). We tested two significant parameters in PPO to find the best performance of our RL agent for different sizes of DMFBs, the number of concurrent environments, and the number of steps for each update.

Figure 5 shows the training rewards for agents with varying number of concurrent environments and number of steps for each update. Here, we show the training rewards for a 20×20 DMFB and a 30×30 DMFB[2]. We observe that the training is not stable when there are only a few concurrent environments. For example, we see that when there are four environments, the performance of the training model (updated every 16 steps) drops significantly after a few training epochs. We also observe that when there are eight environments, no matter what update step-interval is chosen, the performance of the model is consistently better. Similar trends are observed in training for other sizes of DMFBs. Therefore, we chose eight concurrent environments as the PPO setting for model training.

For each training game of DMFB-Env, a random routing task is generated. During droplet routing, DMFB-Env also generates some random concurrent modules to simulate high-throughput bioassay execution. The training processes for different sizes of DMFBs are shown in Figure 6. A training

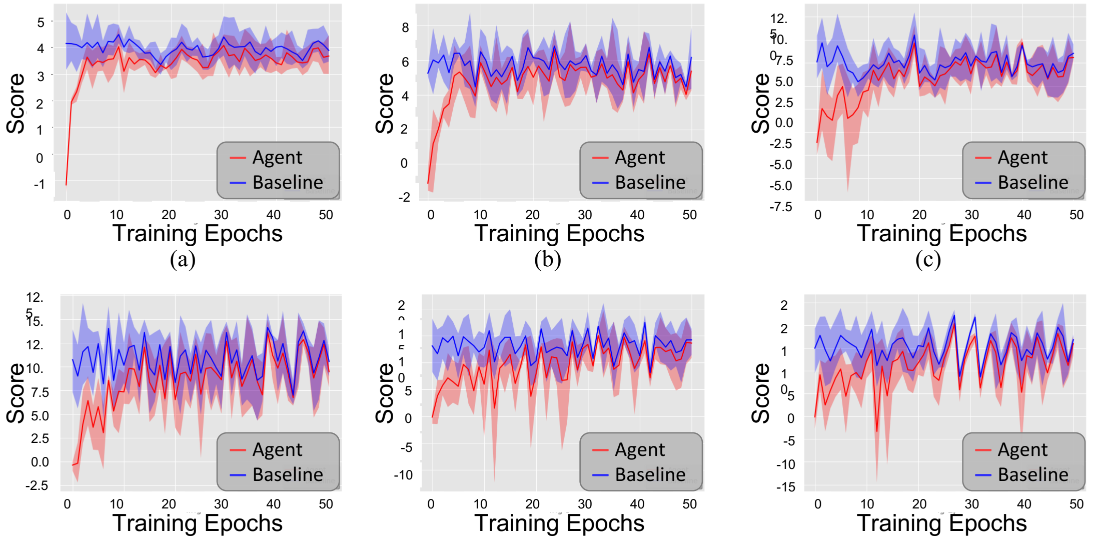
[2] We show the training rewards for other sizes of DMFBs in the supplementary file.



(a)

(b)

Figure 5. Training rewards for agents with different hyper-parameter settings. Score is the total reward that the RL agent receives in a game. (a) Training rewards for DMFBs of size 20×20 electrodes. (b) Training rewards for DMFBs of size 30×30 electrodes.



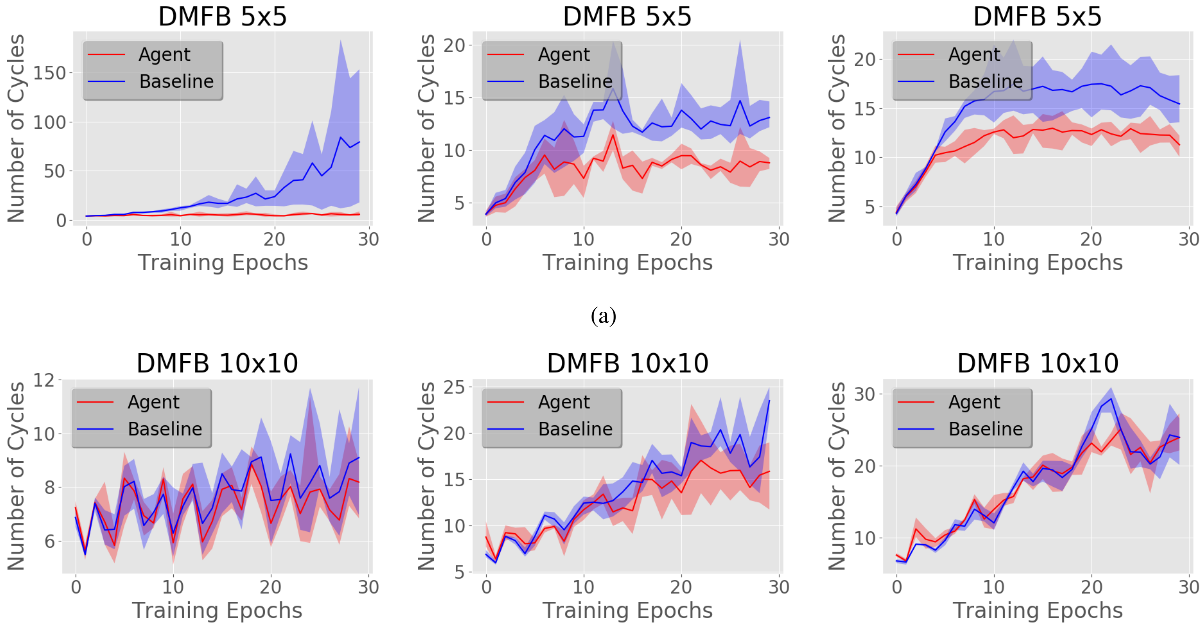
(a) (b) (c)

(d) (e) (f)

Figure 6. Training process corresponding to the models for different sizes of DMFBs. Score is the total reward that the RL agent receives in a game. The performance is compared with a static (offline) routing method in (Zhao & Chakrabarty, 2012). (a) 10×10 DMFB. (b) 15×15 DMFB. (c) 20×20 DMFB. (d) 25×25 DMFB. (e) 30×30 DMFB. (f) 35×35 DMFB.

epoch contains $20,000$ games. From the results, we observe that, for any size of DMFBs, the model is able to achieve the performance of an offline optimization method (Zhao & Chakrabarty, 2012), i.e., the agent is able to transport the droplet to its destination using the shortest path. Intuitively, for larger DMFBs, it takes more epochs to train the model in order to achieve the same performance as the offline method, as the state space is larger for large DMFBs. Although it takes several hours to train a model to perform as well as the offline method, e.g., it takes ~ 4 hours to train a model for the 20×20 DMFB, the training process only needs to be carried out once, and the trained model can be used for all 20×20 DMFBs in the future.

We recorded a video of droplet routing by the agent for a 5×8 DMFB during training, and it can be found in (Liang et al., 2020). From the video we see that, at first, the agent moved the droplet randomly without knowing the right policy needed to reach the destination. After $200K$ training epochs, the agent started to "learn" from past experience; Early on, after $400K$ training epochs, it could transport the



(a)

(b)

Figure 7. Evaluation of the trained models in degrading mode of DMFB-Env. The performance, expressed as the required number of actuation (clock) cycles, is compared with the static routing method from (Zhao & Chakrabarty, 2012). In each sub-figure, three plots (from the left to the right) consider that 10% of the electrodes degrade over time, 50% of the electrodes degrade over time, and 90% of the electrodes degrade over time, respectively. (a) The performance for DMFBs of size 5×5 electrodes. (b) The performance for DMFBs of size 10×10 electrodes.

droplet to the destination using the shortest path for just some of the routing tasks. However, after $800K$ training epochs, the agent could complete all the routing tasks using the shortest paths.

4. Evaluation

To evaluate our RL framework, we considered seven DMFBs with the number of electrodes ranging from 25 to $1,225$. For each DMFB, we first trained three models with the same network architecture (as described in Table 1) using DMFB-Env, and the models were trained in the healthy mode to achieve the same performance as that of the baseline (Zhao & Chakrabarty, 2012). After training, we evaluated the performance of the models in the degrading mode of DMFB-Env. We also evaluated the RL framework by executing an epigenetic bioassay on a fabricated biochip.

4.1. Simulation Results

We compared the performance of the agent with the work in (Zhao & Chakrabarty, 2012). We set different percentages of the degrading electrodes for DMFBs, and the results are shown in Figure 7[3]. Here, we show the number of actuation cycles required in a game as the performance. The fewer actuation cycles required in a game, the better the performance is. We observe that the agent performs similar to the static (offline) method when the DMFBs start to degrade. This is because the RL agent has been trained to perform as well as the baseline in the healthy mode of DMFB-Env. After a small number of training games, the RL agent sometimes performs slightly worse because the agent may explore other alternative routes to avoid the degraded electrodes, and the alternative solutions may be worse than the original route. However, as DMFBs degrade further, the agent can outperform the baseline. We also observe that the proposed solution is more effective for smaller DMFBs. This is because, in our experimental setting, the DMFB with 25 electrodes is the most dynamic environment. The performance of the baseline method decreases if electrode degradation occurs in a DMFB. We see that the performance of the baseline method significantly decreases in the 5×5 DMFB. The experimental results show that the agent can adapt to all sizes of DMFBs, including the most dynamic environment, i.e., the 5×5 DMFB.

We recorded a video of droplet transportation in a degraded environment; the video can be found in (Liang et al., 2020). As some electrodes started to degrade, the agent can still use them to transport the droplet. However, the agent is able to learn the changing health conditions of these electrodes. For subsequent tasks, the agent transports the droplet without using these degraded electrodes.

[3] The complete simulation results for various sizes of DMFBs are provided in the supplementary file.

4.2. Bioassay Execution on a Fabricated Biochip

The RL framework can be used for any bioassay. We designed and executed an epigenetic bioassay on a fabricated DMFB because benchtop epigenetic bioassays require large sample volumes, long execution time, and highly labor intensive. Previous work has shown the effectiveness of epigenetic bioassays on DMFBs (Ibrahim et al., 2016). The executed epigenetic bioassay contains 19 routing tasks, and we used the trained RL droplet router to transport droplets.

4.2.1. Epigenetic Bioassay

Although all the cells in the human body have the same DNA, or genotype, the differences in cell type and function, or phenotype, arise from the selective expression and suppression of certain genes. This phenotypic control is carried out by various epigenetic mechanisms. These are processes and environmental factors that alter genomic behavior and its subsequent expression without any changes to the actual DNA. Epigenetics is the study of these factors and mechanisms of control both in healthy and diseased populations. Chromatin immunoprecipitation (ChIP) is one method of studying the epigenetic relationship between DNA and its supporting proteins (Collas, 2010). Currently, running a full ChIP protocol on a single sample requires a large starting volume of cells (which are not always available), several days to run the assay, and is highly labor intensive. We propose a method of nucleosome immunoprecipitation (NuIP) on magnetic beads as a step towards translating ChIP from the benchtop onto automated DMFBs in the hope of reducing sample sizes, decreasing run times, and increasing throughput.

The NuIP protocol is modified from the traditional ChIP assay (Collas, 2010; Nelson et al., 2006). It consists of first functionalizing a magnetic bead off-chip with an antibody that targets one of the histone proteins in the nucleosome of interest. This is the capture complex as shown in Figure 8. The nucleosome-containing sample is then mixed and incubated with the capture complex followed by magnetic splitting and washing steps. Meanwhile, off-chip, an antibody specific to a different histone protein in the nucleosome is incubated with a fluorescent secondary antibody. This forms the detection complex reagent. Next, the beads are incubated with the detection complex. Should there be any nucleosomes attached to the beads, these will bind with the detection complex. After washing away excess detection complex, ensuring that there are no false positives, the beads are resuspended in a droplet and routed to the detection region. An LED tuned to the excitation wavelength of the fluorescent antibody shines on the beads which are imaged using a CCD camera outfitted with the appropriate emission wavelength filter. A fluorescing sample confirms the presence of the nucleosome of interest.

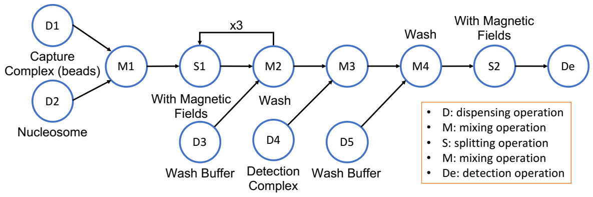
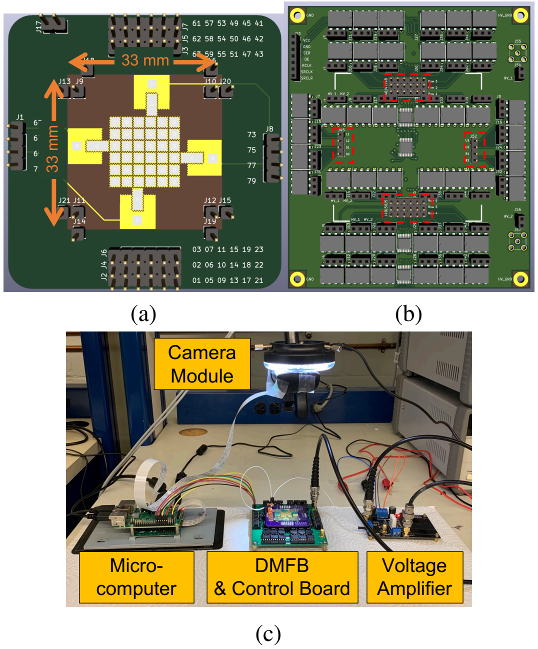


Figure 8. The steps involved in a nucleosome immunoprecipitation assay.

4.2.2. Experimental Setup

Fabricated DMFB: We designed a PCB-based DMFB for the experiment, and fabricated the biochip using services at (OSH, 2020). The DMFB contains a 6 × 6 electrode-array, as shown in Figure 9(a). A reservoir module is placed on each side of the array, and the modules can dispense different reagent droplets. Each electrode can be controlled individually, and these control signals come from the pin heads that are solder on the boundary of the board.

Control board: For the fabricated DMFB, the activation/deactivation status of each electrode is controlled by a high voltage relay (Part No. Panasonic AQW212). As shown in Figure 9(b), a total of 44 relay ICs are soldered on the control board (36 for electrode array and 8 for reservoir modules). Each high-voltage relay IC is controlled by a configuration bit, and these configuration bits are stored in the register ICs (Part No. Texas Instrument SN74AHC595). Besides these ICs, four pin-header modules (shown within



(a) (b)

(c)

Figure 9. (a) The fabricated DMFB. (b) The control board for the DMFB. (c) The experimental setup.

the red rectangles) are used as the DMFB socket, which allows DMFB replacement on the control board.

Overall system: The hardware setup used to operate the digital microfluidic biochip is shown in Figure 9(c). The DMFB is installed above the control board using the pin-header socket. A micro-computer (Part No. Raspberry Pi 3B+) on the left is used to generate control signals to the control board, and the RL agent is installed in the micro-computer. An amplifier board as well as the functional generator are used to generate a voltage source of 1 KHz and 200 Vpp. The voltage source provides actuation signals for the electrodes. A camera module is placed on top of the DMFB to capture droplet locations. The images are utilized by the micro-computer for making real-time decisions.

4.2.3. Experimental Results

We executed the droplet transportation of the bioassay on the fabricated DMFB. To show that the RL agent can adapt to the degrading DMFB environment, we emulated a degraded electrode, the fourth row and the third column electrode, by constantly supplying a lower voltage of 150 Vpp. In the third routing task, the agent used this degraded electrode to transport a droplet, and it experienced the malfunction of the electrode. In the subsequent tasks, the RL agent used other electrodes to transport droplets. We recorded the routing tasks in a video; it can be viewed in (Liang et al., 2020).

5. Discussion

Note that mixing operations are ubiquitous in bio-protocols. On DMFBs, a mixing operation is achieved in two steps: 1) Two droplets are transported to a specific electrode in a mixer module and thus merged into one droplet. 2) The merged droplet is transported in a loop repeatedly until mixing is complete. Therefore, a mixing operation is composed of several routing tasks, and we have considered electrode degradation during mixing. The proposed method makes mixing more reliable when microelectrode degradation occurs in a biochip.

6. Conclusion

We have presented a novel framework for RL-based droplet routing on DMFBs. We have also developed an OpenAI-Gym environment that can be used to train the RL droplet router for various DMFB sizes. The experimental results showed that even though electrodes on a DMFB degrade over time, the RL droplet router can learn the degradation behavior and transport droplets using only healthy electrodes.

We have also demonstrated that using the RL framework, an epigenetic bio-protocol (NuIP) can run much faster and use smaller volumes than with the traditional benchtop protocol. A failure of the DMFB results in costly sample and reagent loss. However, the proposed RL framework minimizes the need to discard biochips with degraded electrodes and abort bioassay protocols. This increases the lifespan of a biochip's utility and allows for the adaptation of a plethora of immunoprecipitation assays on to the DMFB platform.

Acknowledgement

This research was supported in part by the National Science Foundation under grants CCF-1702596 and ECCS-1914796.

References

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.

Chen, Y.-H., Hsu, C.-L., Tsai, L.-C., Huang, T.-W., and Ho, T.-Y. A reliability-oriented placement algorithm for reconfigurable digital microfluidic biochips using 3-d deferred decision making technique. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(8):1151–1162, 2013.

Choi, K., Ng, A. H., Fobel, R., and Wheeler, A. R. Digital microfluidics. *Annual Review of Analytical Chemistry*, 5: 413–440, 2012.

Collas, P. The current state of chromatin immunoprecipitation. *Molecular Biotechnology*, 45(1):87–100, 2010.

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, 2008.

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 (ICRA)*, pp. 3389–3396. IEEE, 2017.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., and Ma, W.-Y. Dual learning for machine translation. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 820–828, 2016.

Ho, T.-Y., Zeng, J., and Chakrabarty, K. Digital microfluidic biochips: A vision for functional diversity and more than moore. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pp. 578–585. IEEE, 2010.

Ho, T.-Y., Chakrabarty, K., and Pop, P. Digital microfluidic biochips: recent research and emerging challenges. In *Proceedings of the IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp. 335–344, 2011.

Hopkins, P. V., Campbell, C., Klug, T., Rogers, S., Raburn-Miller, J., and Kiesling, J. Lysosomal storage disorder screening implementation: Findings from the first six months of full population pilot testing in missouri. *The Journal of Pediatrics*, 166(1):172–177, 2015.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Huang, T.-W., Ho, T.-Y., and Chakrabarty, K. Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pp. 448–455. IEEE, 2011.

Ibrahim, M., Boswell, C., Chakrabarty, K., Scott, K., and Pajic, M. A real-time digital-microfluidic platform for epigenetics. In *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, pp. 1–10, 2016.

Illumina. Illumina NeoPrep Library Prep System. https://emea.illumina.com/company/news-center/press-releases/2015/2018793.html, 2015. [Online; accessed 14-January-2020].

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lehotay, S. J. and Cook, J. M. Sampling and sample processing in pesticide residue analysis. *Journal of Agricultural and Food Chemistry*, 63(18):4395–4404, 2015.

Liang, T.-C., Zhong, Z., Bigdeli, Y., Ho, T.-Y., Chakrabarty, K., and Fair, R. Recorded Videos during Training and Evaluation. https://drive.google.com/drive/folders/1suM0_3GdL1r6_65dTjl_sA78POKi7hRK?usp=sharing, 2020. [Online; accessed 28-June-2020].

Liu, C.-H., Chang, H.-H., Liang, T.-C., and Huang, J.-D. Sample preparation for many-reactant bioassay on dmfbs using common dilution operation sharing. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pp. 615–621. IEEE, 2013.

Luo, Y., Chakrabarty, K., and Ho, T.-Y. Error recovery in cyberphysical digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(1):59–72, 2012.

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 International Conference on Machine Learning*, pp. 1928–1937, 2016.

Myers, F. B. and Lee, L. P. Innovations in optical microfluidic technologies for point-of-care diagnostics. *Lab on a Chip*, 8(12):2015–2031, 2008.

Narasimhan, K., Kulkarni, T., and Barzilay, R. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*, 2015.

Nelson, J. D., Denisenko, O., and Bomsztyk, K. Protocol for the fast chromatin immunoprecipitation (chip) method. *Nature Protocols*, 1(1):179, 2006.

OSH. Park, PCB Fabrication Company. https://oshpark.com, 2020. [Online; accessed 31-January-2020].

Perut, F., Dallari, D., Rani, N., Baldini, N., and Granchi, D. Cell-based assay system for predicting bone regeneration in patient affected by aseptic nonunion and treated with platelet rich fibrin. *Current Pharmaceutical Biotechnology*, 17(12):1079–1088, 2016.

Pierce, V. M. and Hodinka, R. L. Comparison of the genmark diagnostics esensor respiratory viral panel to real-time pcr for detection of respiratory viruses in children. *Journal of Clinical Microbiology*, 50(11):3458–3465, 2012.

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.

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.

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.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Su, F. and Chakrabarty, K. Architectural-level synthesis of digital microfluidics-based biochips. In *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 223–228, 2004.

Su, F. and Chakrabarty, K. Yield enhancement of reconfigurable microfluidics-based biochips using interstitial redundancy. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2(2):104–128, 2006.

Su, F., Chakrabarty, K., and Fair, R. B. Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 25(2):211–223, 2006a.

Su, F., Hwang, W., and Chakrabarty, K. Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of the Design Automation & Test in Europe Conference*, volume 1, pp. 1–6, 2006b.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

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 (ASPLOS)*, pp. 183–197, 2019.

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

Zhao, Y. and Chakrabarty, K. Simultaneous optimization of droplet routing and control-pin mapping to electrodes in digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(2):242–254, 2012.

Zhong, Z., Liang, T.-C., and Chakrabarty, K. Reliability-oriented ieee std. 1687 network design and block-aware high-level synthesis for meda biochips. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 544–549. IEEE, 2020.