Adaptive Droplet Routing for MEDA Biochips via Deep Reinforcement Learning*

Mahmoud Elfar, Tung-Che Liang, Krishnendu Chakrabarty, and Miroslav Pajic
Duke University, Durham NC, USA
{mahmoud.elfar, tung.che.liang, krishnendu.chakrabarty, miroslav.pajic}@duke.edu

***Abstract*—Digital microfluidic biochips (DMFBs) based on a micro-electrode-dot-array (MEDA) architecture provide fine-grained control and sensing of droplets in real-time. However, excessive actuation of microelectrodes in MEDA biochips can lead to charge trapping during bioassay execution, causing the failure of microelectrodes and erroneous bioassay outcomes. A recently proposed enhancement to MEDA allows run-time measurement of microelectrode health information, thereby enabling synthesis of adaptive routing strategies for droplets. However, existing synthesis solutions are computationally infeasible for large MEDA biochips that have been commercialized. In this paper, we propose a synthesis framework for adaptive droplet routing in MEDA biochips via deep reinforcement learning (DRL). The framework utilizes the real-time microelectrode health feedback to synthesize droplet routes that proactively minimize the likelihood of charge trapping. We show how the adaptive routing strategies can be synthesized using DRL. We implement the DRL agent, the MEDA simulation environment, and the bioassay scheduler using the OpenAI Gym environment. Our framework obtains adaptive routing policies efficiently for COVID-19 testing protocols on large arrays that reflect the sizes of commercial MEDA biochips available in the marketplace, significantly increasing probabilities of successful bioassay completion compared to existing methods.**

I. Introduction

Digital microfluidic biochip (DMFB) technology enables the automated manipulation of fluid droplets on the micro-scale. The ability of DMFBs to efficiently execute biochemical protocols has led to its usage in a wide range of applications, including point-of-care clinical diagnostics, biomolecular recognition, air-quality monitoring, and Rapid Acceleration of Diagnostics (RADx) [1], [2], [3], [4], [5]. A recent enhancement to DMFB technology is microelectrode-dot-array (MEDA), which like DMFBs, uses microelectrodes to manipulate droplets based on the electrowetting-on-dielectric (EWOD) principle. In contrast to DMFBs, MEDA biochips are fabricated using TSMC $0.35\,\mu\text{m}$ CMOS technology [6], [7], resulting in relatively smaller microelectrodes that can be dynamically clustered in runtime to form various microfluidic modules such as droplet mixers or splitters. In addition, MEDA biochips offer real-time droplet sensing capabilities via capacitive-sensing at the microelectrode level [8], [9].

Microelectrode degradation is a major concern associated with both DMFBs and MEDA biochip [10]. Specifically, repetitive charging and discharging of a microelectrode, charge trapping, and the degradation of the insulating layer drastically reduces the EWOD force generated by the microelectrode, potentially leading to failures at the bioassay level. Techniques proposed to mitigate the impact of microelectrode degradation on bioassay execution can be classified into reactive and preventive. Reactive techniques aim to recover from errors after their occurrence during bioassay execution; examples include droplet remixing and resplitting [11], dynamic reconfiguration of the biochip [12], [13], and adaptive routing based on the knowledge of the fault map [14]. In contrast, preventive techniques aim to proactively predict and avoid failures. In this category, a recently proposed technique is adaptive routing based on the sensing of microelectrode health levels [15].

Reinforcement learning-based droplet routing has been proposed for adaptive droplet routing in DMFBs [14]. A drawback of this method is that it is reactive, i.e., it detects microelectrode degradation after a fault occurs during runtime and adapts the learned policy based on the fault occurrence. Such reactive adaptation to the relatively rapid degradation of microelectrodes is especially detrimental for applications such as flash chemistry that require fast time to response [16].

Formal synthesis combined with the monitoring of microelectrode health status has been proposed to synthesize droplet routing strategies in MEDA [15]. While such a framework can provide routing strategies with formal (probabilistic) guarantees, the lack of scalability with respect to the biochip size limits the practical applicability of this method. For instance, synthesizing a routing strategy for a mere 30×30 biochip takes an average of nine seconds [15] and is repeated for each microfluidic operation, which is unacceptable for time-sensitive bioassays [16]. In addition, state-of-the-art biochips incorporate a large number of microelectrodes. For example, aQdrop from Sharp Life Science includes 41000 microelectrodes [17], which is at least an order of magnitude larger than what is feasible using the synthesis method described in [15]. With the explosion of the state space and the inefficient storage of strategies, the problem of formally pre-synthesizing droplet routing strategies for such biochips becomes infeasible.

In this paper, we address the problem of designing efficient droplet routing policies for MEDA biochips with proactive mitigation of microelectrode degradation via a deep reinforcement learning (DRL) framework. Our framework uses offline DRL to train a deep neural network (DNN) on a MEDA simulation environment, and online DRL to adapt to the degradation rates of individual biochips. In contrast to

* This research was supported in part by the National Science Foundation under grant No. ECCS-1914796.

formally synthesized strategies [15], the proposed framework efficiently stores droplet routing policies for all possible routing jobs as a DNN, eliminating run-time delays and making it more suitable for time-sensitive bioassays [16]. Since our framework does not require state space enumeration, the DNN adopts a parametrized action space where droplet movements depend on its size, reducing the time required to complete a routing job compared to the usage of single- or double-step movements [14], [15]. Moreover, our framework can be utilized for large MEDA biochips employed in practice [17].

The contributions of this paper are as follows.

- We propose a DRL droplet routing framework for MEDA that exploits feedback about the microelectrode health.
- We develop a stochastic model for MEDA that is suitable for the DRL framework, where we utilize a parametrized action space and adaptive droplet movement.
- We design a DNN specific for the MEDA environment to ensure the scalability of the framework, and we further show training results for various environment configurations.
- We validate the degradation model used in this paper by fabricating PCB prototypes and measuring the degradation of electrodes under voltage actuation.
- We evaluate the proposed framework for COVID-19 testing bioassays and compare this solution to exiting methods.

The rest of this paper is structured as follows. Section II provides background on MEDA biochips, microelectrode degradation and adaptive routing. Section III introduces the proposed online DRL framework for adaptive droplet routing. Section IV provides the DNN training details. Section V compares the performance of the proposed DRL framework to existing methods for adaptive droplet routing, before concluding the paper in Section VI.

II. Background and Motivation

A MEDA biochip manipulates fluids as discrete droplets of picoliter volume using EWOD [18]. Multiple microelectrodes can be dynamically grouped to form a fluidic module (e.g., splitter or mixer) during bioassay execution. A typical MEDA biochip is composed of an array of identical microelectrode cells (MCs). Each MC consists of a microelectrode, an electronic control circuit, and a sensing circuit that enables real-time sensing of droplets. In each operational cycle, the sensing circuit discharges and charges the microelectrode, and measures the charging time. The charging time is used to detect whether a droplet is present over the microelectrode. To obtain the positions of on-chip droplets, the sensing results of all the microelectrodes are shifted out using a scan chain.

A number of failure mechanisms for DMFBs has been identified [19]. Some are related to manufacturing defects; post-fabrication testing can be used to screen for such defects [20], [21]. However, charge trapping in the dielectric layer and degradation of the insulator can result in microelectrode degradation [22], [23]. If an electrode is degraded during bioassay execution, fluidic operations associated with this degraded electrode will fail, resulting in bioassay failure [24], [14].

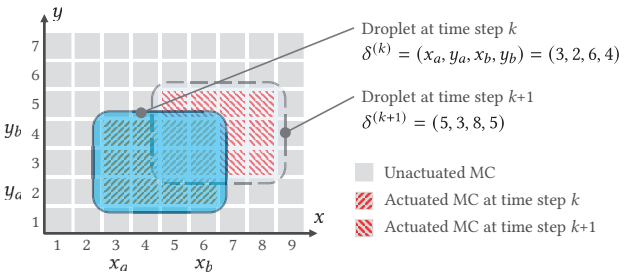

Fig. 1: Example for droplet coordinates at time steps k and $k+1$.

Hence, to reliably execute bioassays on MEDA biochips, we must proactively avoid the use of degraded microelectrodes.

To overcome the problem of electrode degradation in EWOD devices, an adaptive droplet routing method using reinforcement learning was recently proposed [14], employing DRL models to dynamically learn the degradation process in a biochip. However, this approach suffers from a key limitation: in order to provide reliable routing pathways, the model needs to learn from the interaction history of the degraded electrodes. Thus, it is possible that some fluidic operations may fail when the RL agent first encounters the degraded electrode, i.e., the agent has not yet learned the degradation process.

Finally, a new MC design, using the inherent capacitive sensing for microelectrode health monitoring, was recently proposed for MEDA [15], along with effective formal synthesis methods that exploits the health status of microelectrodes.

III. MEDA Modeling for DRL

We now introduce a MEDA biochip model that aids in effective learning of droplet routing strategies. For notation used in the model: $\mathbb{Z}$, $\mathbb{N}_0$ and $\mathbb{R}$ denote the set of integer, natural, and real numbers, respectively. We use $\mathcal{U}\{i, j\}$ and $\mathcal{U}(a, b)$ to denote the discrete (i.e., over integers) and the continuous (i.e., over reals) uniform distributions over $[\![i, j]\!]$ and $[a, b]$, respectively.

MEDA Training Environment: Consider a MEDA biochip of size $\mathsf{W} \times \mathsf{H}$, denoting the number of MCs in each row and column, respectively. Following [15], we model a droplet as a quadruple $\delta = (x_a, y_a, x_b, y_b) \in \Delta$, where $\Delta \subset \mathbb{N}_0^4$ is the set of all possible droplets. A routing task is characterized by the initial (start) and target (goal) droplet locations, denoted by δ_s and δ_g, respectively. We use $\delta^{(k)}$, $k \in \mathbb{N}_0$, to denote the droplet location at the kth control step. Fig. 1 shows an example of the droplet location.

Let (i, j) be the coordinates of a given MC; $D_{ij} \in [0, 1]$ be its degradation level, where 1 indicates a fully healthy MC and 0 a fully degraded; and $n_{ij} \in \mathbb{N}_0$ be the total number of control steps at which the MC was actuated. The degradation level can be estimated as $D_{ij}^{(n)} = \tau_{ij}^{n_{ij}/c_{ij}} \in [0, 1]$, where $c_{ij} \in \mathbb{R}_{>0}$ and $\tau_{ij} \in [0, 1)$ are parameters controlling the degradation rate. Those parameters are generally unknown, although their range can be experimentally estimated [22]. The degradation level of an MC can be measured through the health measurement

Algorithm 1: Procedure for Computing Number of Steps

Input: Droplet δ: (x_a, y_a, x_b, y_b); goal δ_g: $(x_{ag}, y_{ag}, x_{bg}, y_{bg})$; action $a \in \mathcal{A}$
Output: Signed distance (λ_x, λ_y)

$(\Delta x, \Delta y) \leftarrow (x_{ag} - x_a, y_{ag} - y_a)$
$(w_2, h_2) \leftarrow (\lfloor (x_b - x_a + 1)/2 \rfloor, \lfloor (y_b - y_a + 1)/2 \rfloor)$
if $a \in \{a_{\mathtt{N}}, a_{\mathtt{NE}}, a_{\mathtt{NW}}\}$ **then** $\lambda_y \leftarrow h_2 - (h_2 - \Delta y)\mathbb{1}_{\{0<\Delta y<h_2\}}$
if $a \in \{a_{\mathtt{S}}, a_{\mathtt{SE}}, a_{\mathtt{SW}}\}$ **then** $\lambda_y \leftarrow -h_2 + (h_2 + \Delta y)\mathbb{1}_{\{-h_2<\Delta y<0\}}$
if $a \in \{a_{\mathtt{E}}, a_{\mathtt{NE}}, a_{\mathtt{SE}}\}$ **then** $\lambda_x \leftarrow w_2 - (w_2 - \Delta x)\mathbb{1}_{\{0<\Delta x<w_2\}}$
if $a \in \{a_{\mathtt{W}}, a_{\mathtt{NW}}, a_{\mathtt{SW}}\}$ **then** $\lambda_x \leftarrow -w_2 + (w_2 + \Delta x)\mathbb{1}_{\{-w_2<\Delta x<0\}}$

unit [15]. Given a health measurement unit with b-number of bits, the MC measured health is captured by

$$H_{ij}^{(n)} = \lfloor 2^b \cdot D_{ij}^{(n)} \rfloor = \lfloor 2^b \cdot \tau^{n_{ij}/c_{ij}} \rfloor. \tag{1}$$

On the droplet manipulation level, MEDA biochips support single-step movements in the cardinal directions $\mathcal{A}_d$, double-step movements in the cardinal directions $\mathcal{A}_{dd}$, and movements in the ordinal directions $\mathcal{A}_{dd'}$. The action space for droplet manipulation is hence defined as $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_{dd} \cup \mathcal{A}_{dd'}$. The probability that an action is successful largely depends on the health level of the group of microelectrodes — referred to as the *frontier set* — primarily responsible for generating the EWOD force for the action to be performed. We employ the probabilistic transitions modeling from [15]. Each action, along with the current droplet location, determines the group of microelectrodes to be actuated. We use $\mathbf{U} \in \{0,1\}^{\mathsf{W}\times\mathsf{H}}$ to denote the actuation pattern matrix (pattern, for short), where $U_{ij} = 1$ indicates that the microelectrode is actuated.

Parametrized Action Space: In contrast to the traditional action space where each action is associated with an exact number of steps to move, the parameterized action space determines only the direction towards which the droplet is to move, while the number of steps is defined based on the droplet size, shape, and its location relative to the goal. The parametrization of the action space serves multiple purposes. First, it reduces the dimensionality of the model by reducing the action space size. Second, it unifies the action set across different droplet shapes and sizes, enabling the usage of one trained agent for the entire range of droplet sizes. Finally, it allows for moving a droplet beyond two steps at a time.

We define the parameterized action space as the set $\mathcal{A} = \{a_{\mathtt{N}}, a_{\mathtt{S}}, a_{\mathtt{E}}, a_{\mathtt{W}}, a_{\mathtt{NE}}, a_{\mathtt{NW}}, a_{\mathtt{SE}}, a_{\mathtt{SW}}\}$. Let $(\lambda_x, \lambda_y) \in \mathbb{Z}$ be the signed distance (distance, for short) associated with the adaptive action $a \in \mathcal{A}$. Algorithm 1 presents the procedure for computing (λ_x, λ_y) given the current droplet location δ, goal location δ_g, and action a. Basically, the procedure computes the distance based on the droplet size and the movement direction, while avoiding overshooting the goal location. For instance, the droplet shown in Fig. 1 is of size 4×3. Since the maximum reliable distance for the droplet to travel is $(\lambda_x, \lambda_y) = (\lfloor w/2 \rfloor, \lfloor h/2 \rfloor) = (2, 1)$, the adaptive action $a_{\mathtt{NE}}$ attempts to move the droplet one and two steps in the east and north directions, respectively, within the same control cycle.

Observation Space: At each control step k, the DRL agent can observe the current sensor matrix $\mathbf{Y} \in \{0,1\}^{\mathsf{W}\times\mathsf{H}}$ and the health matrix $\mathbf{H}^{(k)}$. For a droplet $\delta = (x_a, y_a, x_b, y_b)$, $Y_{ij} = 1$ for all $(i,j) \in [\![x_a, x_b]\!] \times [\![y_a, y_b]\!]$, while $Y_{ij} = 0$ indicates no droplet is sensed at the indicated MC. In addition, the observation space incorporates the goal location δ_g and the biochip area allocated for routing.

Reward Function: The primary goal in adaptive droplet routing is for the droplet to reach the target location. Performance metrics in this case include the time and distance traveled by the droplet. Since excessive actuations of individual microelectrodes can lead to their premature failure, the number of actuations per microelectrode is to be incorporated in the routing process.

Let $a^{(k)}$ be the action taken at step k from state $s^{(k)}$, resulting in a new state $s^{(k+1)}$. Thus, the reward is defined as $r(k) = \alpha_{\text{dis}} r_{\text{dis}}^{(k)} + \alpha_{\text{ter}} r_{\text{ter}}^{(k)} + \alpha_{\text{act}} r_{\text{act}}^{(k)}$ where r_{dis}, r_{deg} and r_{ter} are the distance, terminal and action rewards, respectively, and $\alpha_i \in \mathbb{R}$ are the respective hyperparameters. To incentivize progression towards the target location, r_{dis} is defined as $r_{\text{dis}}^{(k)} = D\left(\delta^{(k)}, \delta_g\right) - D\left(\delta^{(t+1)}, \delta_g\right)$, where $D\left(\delta^{(k)}, \delta_g\right)$ denotes the Manhattan distance between two droplet locations. The terminal reward r_{ter} aids in faster convergence by associating reaching the target location with an additional reward, defined as $r_{\text{ter}}^{(k)} = \mathbb{1}\{\delta = \delta_g\}$. Finally, the action reward r_{act} penalizes selecting an invalid action, i.e., an action that causes the droplet to exit the routing job area. The selection of the hyperparameters α_i is discussed in Section IV.

IV. DRL Agent Design and Training

A. Training Configurations

We first discuss configuration parameters that affect the training convergence speed – i.e., MEDA biochip size, droplet size, the initial and target droplet locations, the initial microelectrode degradation levels, and the degradation parameters.

Biochip and Droplet Sizes: For training, we considered biochips of sizes between 30×30 and 180×180. We trained the agent for the most common droplet sizes, with droplet width and height $w, h \in \{2, 3, 4, 5, 6\}$, where $w/h \in [0.8, 1.25]$. We assume that the droplet size is preserved throughout a single routing job. Hence, there are two approaches to droplet size selection during training. In the first, multiple agents are utilized, where each agent is trained for a specific droplet size. In the second, the same agent is trained against the range of droplet sizes. Note that a DNN can be feature-invariant by training against the range of values for such feature. Moreover, the exact size of droplets during execution might slightly vary outside those specific values. Consequently, we opt for training a single agent in this framework (i.e., the sedone approach).

Initial and Target Locations: In MEDA biochips, a droplet is either the result of a preceding microfluidic operation or dispensed by an on-chip dispenser. In the former case, the droplet location can be anywhere on the biochip; in the latter, the initial location δ_s is one of multiple, predefined dispenser coordinates. Similarly, the target location δ_g can be either where a microfluidic module is (e.g., a mixer or a splitter), or a predefined exit through one of the biochip reservoirs.

For benchmark bioassays, the percentage of routing jobs involving initial (e.g., dispensing operations) or target (e.g.,

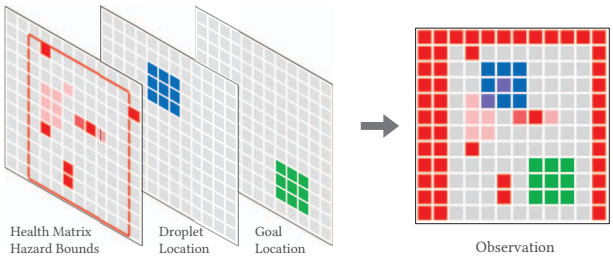

Fig. 2: Channels comprising the observation space.

TABLE I: CNN layers and their configurations.

Layer	Type	Activation	Size	Stride	Padding
L1	Convolution	ReLU	64	3	1
L2	Convolution	ReLU	128	3	1
L3	Convolution	ReLU	128	3	1
L4	Fully-connected	ReLU	256	3	1
L5	Output	ReLU	8	–	–

Algorithm 2: Procedure for Learning Routing Policies

Input: MEDA biochip size
for *epoch* **do**
  $resample \leftarrow \top$
  for $iter = 1, 2, \ldots, N_{iter}$ **do**
    for $actor = 1, 2, \ldots, N_{actor}$, *running in parallel,* **do**
      if $resample = \top$ **then**
        Sample δ_s, δ_g, (τ_{ij}), (c_{ij}), and $\mathbf{N}$
        $resample \leftarrow \bot$
      Run current policy π and obtain rewards and new state
      if $(k \geq k_{\max}) \vee (\delta = \delta_g)$ **then** $resample \leftarrow \top$
    if $(iter \cdot N_{actor}) \bmod minibatchsize = 0$ **then**
      Optimize PPO2 loss function, update current policy π

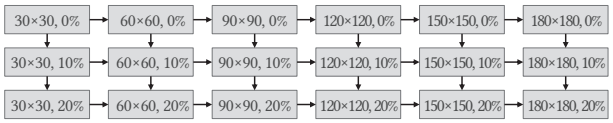

Fig. 3: Dataflow diagram for transfer learning. An arrow indicates that the source CNN is used to initialize the training of the destination CNN.

discarding operations) droplets adjacent to one of the biochip edges is between 20% and 40% [25]. Thus, during training both the initial and goal locations are sampled from a stratified distribution. Specifically, we randomly sample δ_s and δ_g at the start of each training episode such that $x_{as}, x_{ag} \sim \mathcal{U}\{2, \mathsf{W}{-}w{-}1\}$, and $y_{as}, y_{ag} \sim \mathcal{U}\{2, \mathsf{H}{-}h{-}1\}$.

Degradation Parameters: From (1), degradation parameters of microelectrodes affect their degradation rate, although they are not directly observable to the agent. For training, we randomly sample the degradation parameters as $c_{ij} \sim \mathcal{U}(c_{\min}, c_{\max})$ and $\tau_{ij} \sim \mathcal{U}(\tau_{\min}, \tau_{\max})$, where the distributions are experimentally obtained as described in Section V. On the other hand, the number actuations n_{ij} is updated based on the actuation patterns applied by the agent at each step.

B. DNN Design and Training

We employ a convolutional neural network (CNN) to learn droplet routing policies due to its potential in preserving important features of the observation space. As illustrated in Fig. 2, the input to the CNN is a matrix of size $(\mathsf{H}, \mathsf{W}, 3)$. The three channels represent the microelectrode health levels and routing zone, the goal location, the current droplet location. The agent's goal is to learn a policy that maximized the expected cumulative reward.

For training, we use the proximal policy optimization (PPO) algorithm [26], [27] with actor-critic architecture. Unlike policy gradient methods for reinforcement learning where policy gradients are updated after reading each data sample, PPO utilizes a surrogate objective to stabilize the training process via multiple workers. Algorithm 2 summarizes the procedure for training the agent to learn droplet routing policies. Each training episode starts with the initial state $(\delta_g, \mathbf{H})$, sampled according to the distributions described earlier. After each step, the number of actuations n_{ij} is updated for all MCs using the actuation pattern matrix $\mathbf{U}$. An episode is terminated when either the target location is reached, i.e., $\delta = \delta_g$, or the maximum number of steps allowed has passed, i.e., $k = 2(\mathsf{W} + \mathsf{H})$. Using the accumulated rewards, the gradients for each encountered (s, a) are updated.

To avoid catastrophic unlearning, we adopt a dynamic learning rate scheduler for training. Specifically, the training process starts with a base learning rate η_0. At the end of the ith epoch, the learning rate is discounted with factor β_η only if the agent performance is above a certain threshold, i.e.,

$$\eta_{i+1} = \begin{cases} \max\left(\beta_\eta * \eta_i, \eta_{\min}\right) & \text{success rate} > 0.99, \\ \eta_i & \text{otherwise.} \end{cases}$$

Through hyperparameter optimization, we chose $\eta_0 = 3.5 \times 10^{-4}$, $\eta_{\min} = 1.0 \times 10^{-6}$, and $\beta_\eta = 0.7$.

To accelerate the training process, we apply transfer learning as follows. We first train a randomly initialized CNN on biochips of size 30×30 with no fault injection. Next, we use the pre-trained base CNN to initialize the training of the CNNs used for the next biochip size and fault injection level, and the process is repeated using the new CNNs as shown in Fig. 3. To facilitate the transfer learning, the input layer size is unified across all CNNs by scaling the observation matrix from the original size, i.e., $(\mathsf{W}, \mathsf{H}, 3)$, to $(30, 30, 3)$. A comparison of the training performance between random initialization and transfer learning approaches is discussed in Section V.

V. Experimental Evaluation

A. Estimation of Degradation Parameters

The first set of experiments aims to validate the degradation model and to estimate the parameters in (1). To this end, we monitored electrode degradation in several identical PCB-based DMFBs. Note that these biochips manipulate droplets using the same EWOD principle as MEDA biochips. The DMFBs contain electrodes in three sizes: $2{\times}2\,\mathrm{mm}^2$, $3{\times}3\,\mathrm{mm}^2$, and $4{\times}4\,\mathrm{mm}^2$ (see Fig. 4). Four reservoir modules are placed on two sides of the biochip, and these modules can dispense different reagent droplets. Each electrode can be controlled individually, and these control signals come from

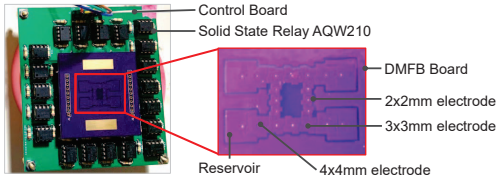

Fig. 4: The experimental setup.

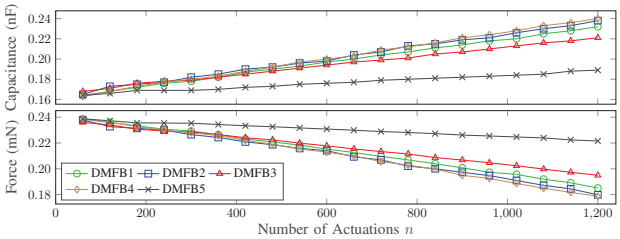

Fig. 5: Capacitance increase (top) and EWOD force degradation (bottom).

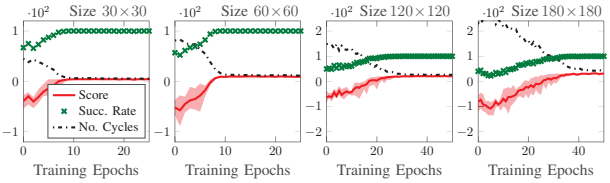

Fig. 6: Performance results for training CNNs on healthy MEDA biochips.

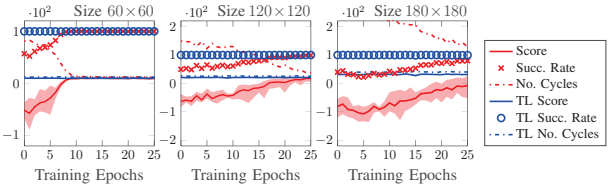

Fig. 7: Performance results for training CNNs via random initialization (red) and transfer learning (blue).

a control board underneath the DMFB. The activation/de-activation status of each electrode is controlled by a high voltage relay. Each high-voltage relay IC is controlled by a configuration bit, and these configuration bits are stored in the register ICs. The hardware setup used to operate the digital microfluidic biochip is shown in Fig. 4. To efficiently run the experiments, we executed the same actuation pattern on two DMFBs at the same time.

We developed an actuation sequence for the electrodes that leads to repeated fluidic operations on the biochip. When executing the actuation sequence, each electrode is actuated for 1 s for hundreds of times. After executing the sequence, we actuated an electrode and measured the charging times needed using an oscilloscope. Because the electrode and the top plate form a capacitor, and a resistor is placed in series with the electrode, the charging path is a simple RC circuit. The effective capacitance of an electrode at time t can then be derived using $V_C(t) = V_{pp}\left(1 - e^{-t/RC}\right)$, where $V_C(t)$ is the electrode capacitance at time t. Subsequently, the EWOD force F can be obtained from [28], [29]

$$F = \frac{C(V_C - V_T)^2}{2}\frac{dA(x)}{dx},$$

where $V_T = 130\text{V}$ is threshold voltage due to soldermask insulator [29], $A(x)$ is the area of the droplet over the activated electrode, and x is the droplet position.

The degradation results, i.e., increase in capacitance and decrease in EWOD force, of five DMFBs are shown in Fig. 5 — the capacitance of an electrode grows linearly as we repeatedly actuate the electrode, decreasing the EWOD force. The degradation parameters are estimated as $\tau \in [0.5, 0.7]$ and $c \in [500, 800]$, and are further used for DRL agent training.

B. Evaluation

We next present the results for training agents for various configurations by showing the mean score of the agents after each training epochs for biochips of sizes $\mathsf{W} \times \mathsf{H}$, where $\mathsf{W} = \mathsf{H} \in \{30, 60, 120, 180\}$. Performance metrics consist of the mean score, the number of cycles, as well as the success rate. The metrics are collected after each training epoch by testing the agent for 500 random routing jobs, and each experiment is repeated five times. All experiments were carried out with eight parallel environments and 2^{14} total number of steps. The training and experiments were conducted on an Intel Xeon Silver 4208 CPU and an Nvidia RTX 6000 GPU with 24 GB of memory. The training and simulation environment were implemented using Python, including OpenAI Gym and Stable-Baselines libraries.

We first trained the CNN on a healthy MEDA biochip, i.e., the number of actuations per each microelectrode is reset at the beginning of each training episode. Fig. 6 presents the CNN performance metrics versus the number of training epochs. The trends show that after a low number of epochs, a CNN learns an effective policy — i.e., the success rate converges to 100% and the score and the average number of cycles stabilize — at a relatively small number of epochs that ranges from 10 to 40 and increases with the biochip size.

We then deployed transfer learning by using the CNN trained on healthy MEDA biochips of size 30×30 to initialize the training of the CNN for the next biochip size, i.e., 60×60. The process was repeated subsequently for the remaining biochip sizes. Fig. 7 compares the performance of the CNNs trained via random initialization against the ones trained via transfer learning. For all biochip sizes, the transfer learning-based CNNs were able to learn effective policies within the first training epoch, exhibiting the same performance that the CNNs trained via random initialization were able to achieve after 15 to 40 epochs.

We also tested the robustness of the trained agents against randomly injected faults at runtime. We used the agents trained on healthy biochips to initialize the training against biochips with randomly injected faults. Before each training episode, a fixed percentage of fully-degraded microelectrodes are randomly placed in clusters of size 2×2. Similar to the

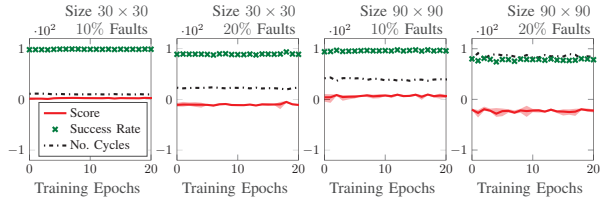

Fig. 8: Performance results for transfer learning in fault injection.

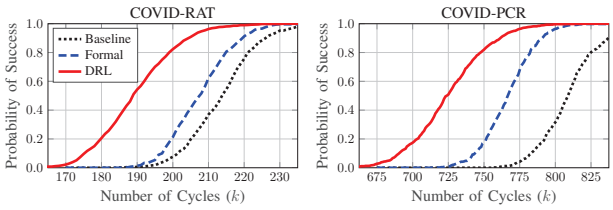

Fig. 9: Probability of successful bioassay completion vs. no. of cycles.

previous experiments, the trained agents were used to initialize the training on a higher percentage of faults. Fig. 8 shows the performance results for training against 10% and 20% fault injection modes. The trends demonstrate that the agents were able to adapt to the faults within the first training epoch.

Finally, to evaluate the trained CNNs, we run experiments where we compare their performance against two baselines: (i) health-agnostic policies that aim to minimize the time to reach the target without knowledge of the MC health levels (baseline), and (ii) formally synthesized strategies using PRISM-games model checker (formal) [15]. Each policy was used to execute two benchmark bioassays that are used for COVID-19 testing: PCR-based (COVID-PCR) and rapid antigen-based (COVID-RAT), are widely used to detect the presence of the SARS-CoV-2 virus or the body's response to infection [25].

Fig. 9 shows the probability of successful bioassay completion within a given number of cycles k. The graph shows that the DRL outperforms the policies from the literature by achieving a significantly higher probability of success. The gain in performance is primarily due to the utilization of adaptive movement distance (see Section III). For instance, DRL successfully executed COVID-PCR within $k = 762$ with probability $p > 0.9$, compared to $p < 0.4$ using the other policies. In addition, the time needed to obtain a routing policy from the trained CNN is negligible ($t < 0.1$ sec) when compared to the formally synthesized policies where t ranged from 5 to 48 sec before each routing job.

VI. Conclusion

In this paper, we have proposed a deep reinforcement learning (DRL) framework to address the problem of designing droplet routing policies for MEDA biochips with proactive mitigation of microelectrode degradation. The proposed framework utilizes the recently-developed technology of microelectrode health monitoring by incorporating the microelectrode health status into the observation space. Our framework uses DRL to train CNNs for droplet routing policies in MEDA biochips. We have shown that the proposed framework is superior to existing formal synthesis techniques in terms of the probability of successful bioassay completion and the scalability with respect to the biochip size.

References

[1] W.-L. Chou *et al.*, "Recent advances in applications of droplet microfluidics," *Micromachines*, vol. 6, no.9, pp.1249–1271, 2015.

[2] R. S. Sista *et al.*, "Digital microfluidic platform to maximize diagnostic tests with low sample volumes from newborns and pediatric patients," *Diagnostics*, vol. 10, no.1, p.21, 2020.

[3] S. Huang *et al.*, "Digital microfluidics for the detection of selected inorganic ions in aerosols," *Sensors*, vol. 20, no.5, p.1281, 2020.

[4] A. Ganguli *et al.*, "Rapid isothermal amplification and portable detection system for SARS-CoV-2," *Proc. NAS*, 2020.

[5] C. Sheridan, "COVID-19 spurs wave of innovative diagnostics," *Nature biotechnology*, vol. 38, no.7, pp.769–772, 2020.

[6] K. Y.-T. Lai, Y.-T. Yang, and C.-Y. Lee, "An intelligent digital microfluidic processor for biomedical detection," *Journal of Signal Processing Systems*, vol. 78, no.1, pp.85–93, 2015.

[7] Y. Ho *et al.*, "Design of a micro-electrode cell for programmable lab-on-cmos platform," in *Proc. ISCAS*, 2016, pp.2871–2874.

[8] S. Poddar *et al.*, "Error-correcting sample preparation with cyberphysical digital microfluidic lab-on-chip," *ACM TODAES*, vol. 22, no.1, pp.1–29, 2016.

[9] T.-C. Liang *et al.*, "Extending the lifetime of MEDA biochips by selective sensing on microelectrodes," *IEEE TCAD*, vol. 39, no.11, pp. 3531–3543, 2020.

[10] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges," *IEEE TCAD*, vol. 25, pp.211–223, 2006.

[11] Z. Zhong, Z. Li, and K. Chakrabarty, "Adaptive and roll-forward error recovery in MEDA biochips based on droplet-aliquot operations and predictive analysis," *IEEE TMSCS*, vol. 4, pp.577–592, 2018.

[12] Z. Li *et al.*, "Error recovery in a micro-electrode-dot-array digital microfluidic biochip," in *Proc. ICCAD*, 2016, pp.1–8.

[13] ——, "Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip," *IEEE TCAD*, vol. 37, pp.601–614, 2017.

[14] T.-C. Liang *et al.*, "Adaptive droplet routing in digital microfluidic biochips using deep reinforcement learning," in *Proc. ICML*, 2020.

[15] M. Elfar *et al.*, "Formal synthesis of adaptive droplet routing for MEDA biochips," in *Proc. DATE*, 2021, pp.324–329.

[16] J.-i. Yoshida, A. Nagaki, and T. Yamada, "Flash chemistry: fast chemical synthesis by using microreactors," *Chemistry–A European Journal*, vol. 14, no.25, pp.7450–7459, 2008.

[17] S. Anderson, B. Hadwen, and C. Brown, "Thin-film-transistor digital microfluidics for high value in vitro diagnostics at the point of need," *Lab on a Chip*, vol. 21, no.5, pp.962–975, 2021.

[18] C. Quilliet and B. Berge, "Electrowetting: a recent outbreak," *Current Opinion in Colloid & Interface Science*, vol. 6, no.1, pp.34–39, 2001.

[19] T. Xu and K. Chakrabarty, "Fault modeling and functional test methods for digital microfluidic biochips," *IEEE TBioCAS*, vol. 3, pp.241–253, 2009.

[20] F. Su, S. Ozev, and K. Chakrabarty, "Ensuring the operational health of droplet-based microelectrofluidic biosensor systems," *IEEE Sensors Journal*, vol. 5, no.4, pp.763–773, 2005.

[21] T. A. Dinh *et al.*, "A general testing method for digital microfluidic biochips under physical constraints," in *2015 IEEE International Test Conference (ITC)*. IEEE, 2015, pp.1–8.

[22] H. Verheijen and M. Prins, "Reversible electrowetting and trapping of charge: model and experiments," *Langmuir*, vol.15, pp.6616–6620, 1999.

[23] A. I. Drygiannakis, A. G. Papathanasiou, and A. G. Boudouvis, "On the connection between dielectric breakdown strength, trapping of charge, and contact angle saturation in electrowetting," *Langmuir*, vol. 25, pp. 147–152, 2009.

[24] Z. Zhong *et al.*, "Micro-electrode-dot-array digital microfluidic biochips: Technology, design automation, and test techniques," *IEEE TBioCAS*, vol. 13, no.2, pp.292–313, 2018.

[25] G. Guglielmi, "Fast coronavirus tests: what they can and can't do." *Nature*, vol. 585, no.7826, pp.496–498, 2020.

[26] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. ICML*, 2016, pp.1928–1937.

[27] J. Schulman *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[28] Z. Li *et al.*, "Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips," *IEEE TBioCAS*, vol. 11, no.3, pp.612–626, 2017.

[29] R. B. Fair, "Digital microfluidics: is a true lab-on-a-chip possible?" *Microfluidics and Nanofluidics*, vol. 3, no.3, pp.245–281, 2007.