– Independent Work Report Fall 2024 –

Predictive Modeling of Optogenetic Cell Migration with Hierarchical Reinforcement Learning

Alice Hou
Adviser: Benjamin Eysenbach

Abstract

OptoEGFR is a novel tool that leverages optogenetics to control cell behaviors with light, which serves as a highly precise and noninvasive stimuli. However, applying this innovation to real-life use cases is currently limited by the complexity of cellular dynamics, which makes it difficult to effectively predict and control cell movement. In this paper, we will present a hierarchical unsupervised RL controller built on top of the JaxGCRL framework that can dynamically generate optimal illumination patterns to guide collective cell movement to form specific goal configurations. This controller operates in real time, leveraging feedback from the environment to iteratively discover the best illumination patterns to drive cellular behaviors such as clustering or adjusting densities in specific regions to shape tissues into their desired configurations. We show that the controller can achieve a high success rate, demonstrating that the controller explores a diverse variety of strategies and applies optimal strategies to produce the final illumination patterns. This has key implications in bridging the current gap between OptoEGFR and its practical potential in vital fields like tissue engineering and regenerative medicine.

1. Introduction

Collective cell migration is a phenomenon that underpins key cellular processes like wound healing, cancer metastasis, and tissue regeneration [24]. Cells naturally coordinate collective movement to repair wounds or in cancer metastasis, to invade surrounding tissue [17]. Being able to precisely control this process opens up transformative new possibilities in regenerative medicine and synthetic tissue engineering. For instance, wound regeneration could be enhanced with the ability to guide

cells to repair damaged tissue regions and engineer specific tissue architectures for transplantation [20]. However, not only does this demand a high degree of spatial and temporal precision, but it also requires the ability to dynamically adapt and respond to complex, constantly changing biological environments.

Traditional methods of controlling collective cell migration include using chemical gradients, mechanical patterns, or electrical signals. However, these approaches are often limited in both precision and efficiency [16, 18, 34]. Optogenetics—a technique that controls cells with light—poses a promising alternative [5]. This method has been used to successfully guide single-cell movement by activating key pathways in cell movement dynamics like Rac1 or PI 3-kinase, demonstrating the ability of optogenetics to foster efficient, specific, and non-invasive control over cell behaviors [35, 30, 8]. However, in scaling to collective cell migration, the added complexity of intercellular interactions poses a new challenge.

OptoEGFR, a light-activated receptor tyrosine kinase (RTK), has unlocked the ability to control collective cell migration. By using light in place of a ligand to activate the RTK, OptoEGFR enables specific and programmable control over the PI 3-kinase pathway, which plays an active role in directing cell migration [26]. Suh et al. demonstrated that light stimulation of OptoEGFR can drive large-scale tissue rearrangements to form specific shapes, such as localized densification of interior areas and directional outgrowth at tissue edges [28].

However, predicting and designing optimal illumination patterns to attain goal cell configurations remains a major challenge due to the variable nature of cell conditions. Cellular responses to light are influenced by factors like tissue geometry, initial cell densities, and local signaling dynamics, which all contribute to the multifaceted nature of this problem [28]. Thus, the goal of this project is to develop a hierarchical controller using unsupervised reinforcement learning (RL) to dynamically create illumination patterns that optimize collective migration to form specific goal configurations. The controller leverages real-time feedback to discover illumination patterns that effectively drive cell behaviors such as clustering or adjusting densities in targeted regions to morph tissues into desired arrangements. By optimizing optogenetic control of cell migration and enabling constant

adaptation based on dynamic feedback from the environment, this project aims to bridge the gap between OptoEGFR's current abilities and its practical potential in crucial applications like tissue engineering and regenerative medicine.

In this paper, we will first provide an overview of the current problem of controlling cell migration and the techniques that have been used thus far. We will then introduce using ML as a solution to the common pitfalls of modeling the complex dynamics of collective cell migration. Next, we will propose a novel hierarchical controller developed using unsupervised RL and built on a framework established by the JaxGCRL codebase. Finally, we will delve into the performance of the controller in guiding the generation of optimal illumination patterns to achieve given goal states.

2. Overview of Challenge and Previous Work

In a 1979 *Scientific American* article, Francis Crick identified a major challenge facing the field of neuroscience: discovering a method to control a specific type of cells in the brain without disturbing others [4]. Over 30 years later, Deisseroth pioneered the field of optogenetics. Optogenetics is a powerful technology that combines genetics and optics to control cellular behaviors with a high degree of spatial and temporal precision [5]. Following Deisseroth's initial discovery, Wu et al. engineered a Rac1 system that induced the actin cytoskeleton—a structure responsible for cell movement and structure—dynamics in localized regions to guide single-cell movement [35]. This demonstrated the early potential of using optogenetics to exercise precise control over single-cell movement.

Toettcher et al. made further advancements by developing a set of tools that can be used to control specific signaling proteins in cells [31]. More specifically, Toettcher et al. used the phytochrome B (Phy)-PIF light-gated protein system to activate the Ras/Erk pathway, which upon activation by extracellular signals, is responsible for processes like cell proliferation or differentiation [1, 19]. Due to the flexibility offered by using light as a stimulus, the researchers were given a high degree of control over the timing and intensity of the stimuli to unearth specific details on the Ras/Erk pathway's filtering capabilities [31]. Ultimately, this work further testifies to how light can be used

as a highly precise stimulus to directly influence signaling dynamics.

Demonstrated by the research of Wu et al. and Toettcher et al., optogenetic control has been highly successful in applications with single-cell migration. However, broadening this to collective migration poses an entirely new set of challenges. Coordinating collective movement must take into consideration additional mechanical forces, signaling dynamics, and intercellular interactions, which all lend to emergent behaviors that are often difficult to predict or capture in a model [17]. Traditional approaches include using chemical gradients, mechanical patterns, or electrical signals, but these methods currently fall short in achieving high precision and efficiency [16, 18, 34]. Chemical methods rely on adjusting concentrations of chemotactic signals to control cell movement, which does not offer much flexibility in being able to dynamically adjust the stimuli once it has already been introduced into the system [16]. Mechanical methods that target durotaxis—a phenomenon where cells tend to move to regions of higher stiffness in their environments—are limited by the stiffness of substrates, and while topological cues—such as grooves, ridges, or other surface-level features—can drive movement, they are not able to achieve a high degree of spatial precision [18]. Finally, while electrical signals have shown potential in guiding cell movement, they often require external electrodes that are not able to localize signals effectively because they impact all cells at the application site without being able to distinguish between different types [34]. Thus, while traditional methods can be used to control collective cell migration, they currently lack the degree of precision and adaptability necessitated by applications such as tissue repair, where the environment is also both highly complex and constantly changing [36].

OptoEGFR, a light-activated receptor tyrosine kinase (RTK), poses a significant advancement in the ability to control collective cell migration. RTKs naturally facilitate collective cell migration and are typically triggered upon binding with a specific ligand [33]. By replacing the ligand with a light stimuli, OptoEGFR enables specific and programmable control over the PI 3-kinase pathway, which plays an active role in directing cell migration [26]. More precisely, activation of the PI 3-kinase pathway has been experimentally shown to be necessary in large-scale tissue movements caused by physical cell movement (rather than ligand gradients), which is targeted

by light stimuli [28, 30, 27]. Suh et al. demonstrated the potential of OptoEGFR in directing the large-scale migration of mammalian cells by using light alone. Depending on the shape of the illumination pattern applied and the initial conditions of the tissue structure, OptoEGFR stimulation resulted in a diverse set of outcomes. For instance, when light was applied to a localized interior region, tissue densification was observed. When light was applied to the outer edges of the cellular layer, tissue outgrowth was observed at speeds approximately 40% faster than control experiments that did not receive any illumination [28]. As demonstrated in Figure 1, when illumination was applied in specific patterns, OptoEGFR cell movement drives tissue rearrangement to conform to the corresponding shape [28].

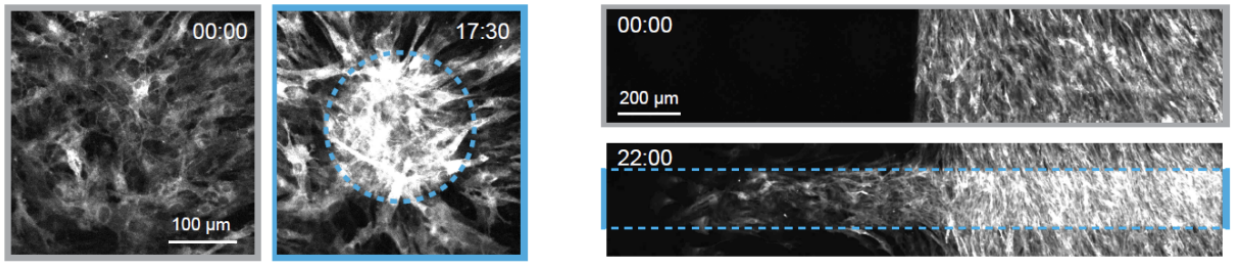


Figure 1: Experiments conducted by Suh et al. illustrate the tissue arrangements of two sets of OptoEGFR cells before and after applying an illumination pattern. The figure on the left shows the effect of applying a circular pattern (shown by the dotted line) and the figure on the right shows the effect of applying a rectangular pattern (shown by the dotted line). The cells migrate from their initial configurations to morph into an approximation of the applied illumination pattern [28].

OptoEGFR allows for precise control of collective cell migration using external illumination patterns, but designing these patterns remains a challenge due to the inherent variability in initial cell conditions and the dynamic nature of cellular responses and tissue geometries [28]. Thus, predicting optimal patterns of light stimuli to achieve desired cellular configurations remains an open challenge.

To address the issue of modeling highly complex cellular environments, researchers have turned to machine learning and methods rooted in RL. For example, Hou et al. developed an agent-based simulation framework in combination with deep RL to optimize the timing of abstract signals in guiding collective cell movement. They discovered that more frequent signals led to accelerated migration, which highlights the potential of RL in applications like optimizing control strategies

[13]. However, their work used generalized stimuli as proxies for biological signals, which limits its direct applicability to real-world systems. Similarly, Zhang et al. used a deep deterministic policy gradient (DDPG) algorithm with leader and follower agents to model collective cell migration. Leaders were trained to find the shortest path to a target, and followers were trained to coordinate motion along local chemical gradients. This model was able to successfully follow migration behaviors that were similar to experimental observations and also outperformed traditional rule-based approaches in migration efficiency and adaptability, which demonstrates the ability of RL to capture complex biological dynamics [37]. LaChance et al. further exemplified the applicability of RL in collective cell migration by developing neural networks that could learn cell coordination rules based on experimental data on cell movement trajectories. The model's ability to mimic behaviors like collective movement and clustering provided valuable insights into the mechanisms underlying intercellular interactions during migration [14]. These studies highlight the effectiveness of RL in modeling complex processes like collective cell migration. However, prior work primarily focuses on simulation rather than active control. As such, there remains a gap that needs to be addressed to bring the current capabilities of OptoEGFR into more practical applications, which lends to the aim of our project to learn and predict optimal light patterns to guide collective cell migration dynamically.

3. Approach

To tackle the problem of collective cell migration, we considered 3 main approaches: supervised reinforcement learning, traditional control theory, and unsupervised reinforcement learning. We will provide an overview of each and rationale for unsupervised RL as the best suited approach.

3.1. Supervised Reinforcement Learning

Supervised RL refers to the setting where an agent interacts with the environment in a trial-and-error manner and uses labeled and/or expert data to guide the learning process. Supervised signals help the agent derive a policy, ultimately reducing the frequency of inefficient or random exploration.

Rewards from the environment influence this process by providing feedback on the agent's actions and simultaneously encouraging exploration of new approaches [25]. However, because OptoEGFR is a relatively new (as of 2024 by Suh et al.), there is a limited amount of data available due to time and resource constraints on experimentation [28]. We reached out to the authors and were provided with a dataset of illumination patterns and cell migration time lapses, but there were only around ~ 20 total experiments captured within the dataset, which was not sufficient for training a large model. Nevertheless, the provided dataset was able to shed light on the specific movement dynamics of the cells and the types of illumination patterns that can be applied. Furthermore, given any goal state, there are potentially infinite possibilities for initial cell configurations and possible illumination patterns that can be generated, which further challenges the reasonability of gathering this data experimentally.

Furthermore, supervised RL agents rely on the provided labeled/expert data, which poses a potential limit to their ability to adapt to new, unseen conditions [25]. The problem of controlling collective cell migration requires fast, real-time adaptability due to the highly variable nature of tissue geometries, cell densities, and signaling pathway responses. This dependency on labeled data makes it challenging for an supervised RL agent to effectively mitigate this dynamic environment. As such, we did not consider supervised RL as the best approach.

3.2. Control Theory

Control theory is a framework that regulates systems by adjusting inputs to achieve specific desired outputs. This method relies on monitoring the state of the system over time and iteratively implementing changes to mitigate any deviations from the desired state [21]. However, traditional control theory is often reliant on linear assumptions of system dynamics and is therefore commonly used in environments that follow fixed and predictable conditions [22, 6]. This is not well suited for the problem statement, as cellular environments are complex and constantly changing, so any potential solution must be able to effectively process and respond to this dynamic environment.

3.3. Unsupervised Reinforcement Learning

Controlling optogenetic cell migration is a spatio-temporal control problem, where we use specific illumination patterns to guide cell movement to form target states. Currently, these patterns are manually designed based on the target states of the cells without taking into consideration the myriad of different initial states the cells can start in. By contrast, our approach leverages hierarchical unsupervised RL with to facilitate real-time adjustments to the illumination pattern based on the current state of the cells.

Unsupervised RL refers to the approach in which agents learn skills without receiving any external reward signals. The agent explores the environment, often relying on intrinsic measures such as diversity or novelty to discover behaviors [7, 11]. For collective cell migration, a reward function would be difficult to define—there is a high degree of variation across cell densities, tissue geometries, and signaling dynamics that may cause a predefined reward function to be sufficient for one initial state, but not for another. For example, an illumination pattern that can achieve clustering in an environment of dense tissue would be different from the pattern needed to achieve the same result in sparse tissue. Unsupervised RL is well suited for situations where it is difficult to define an explicit reward, shifting the dependency to more intrinsic strategies that encourage exploration [7, 15]. Furthermore, a focus on diversity or novelty for intrinsic rewards encourages the controller to discover novel illumination patterns to achieve the desired goal states even without a predefined reward function.

To implement this, we need a framework that can handle high computational demands. Furthermore, we must also define intrinsic rewards and ensure that the system can scale to high-dimensional state spaces that represent initial cell densities. Goal-conditioned RL (GCRL) is a setting in which tasks in the environment are defined by goal states instead of a reward function—for instance, in a maze environment, we could define several goal states (end of the maze or a specific location) and train the agent to learn how to reach these states [9]. JaxGCRL, a GPU-accelerated codebase for self-supervised goal-conditioned RL research, is an ideal foundation for this project. JaxGCRL

relies on the JAX framework, which performs JIT (Just-in-Time) compilation to enable rapid policy updates, efficient gradient calculations, and ease in scaling to complex, high-dimensional environments [2]. These features are particularly useful because the environment is constantly changing, and the agent must respond dynamically to changes in tissue shapes or cell densities.

The architecture of our hierarchical controller has two main layers: the high-level planner and the low-level controller. These layers work together to guide the cells to form a specific configuration that is as close to the desired goal state as possible. In other words, the environment is modeled as a Markov Decision Process (MDP) that separates the control decisions between two time scales: one scale for planning and another that responds by making adjustments.

3.3.1. High-Level Planner

The high-level planner takes in the initial densities of the cells and the target densities according to the goal state to output an approximate illumination intensity for each quadrant of the grid—this forms a starting point for the illumination pattern.

3.3.2. Low-Level Controller

While the planner selects an initial direction, the low-level controller works at a smaller, more specific scale to evaluate how closely the current cell densities mirror the short-term goals proposed by the high-level planner. The controller will continually provide feedback to adjust the illumination pattern to mitigate any discrepancies on a smaller scale.

4. Implementation

Figure 2 shows a schematic diagram of the system we will use to build the hierarchical controller. The actor serves as the controller, which takes as input the cell densities in the current state of the environment. Based on this, the actor generates an illumination pattern to apply to the cells in order to achieve a specific goal configuration. The environment then simulates cell movement in response to the applied illumination, and then outputs the next state, an intrinsic reward signal, and a done flag. These are stored into the replay buffer, which is also sampled from to train the critic network.

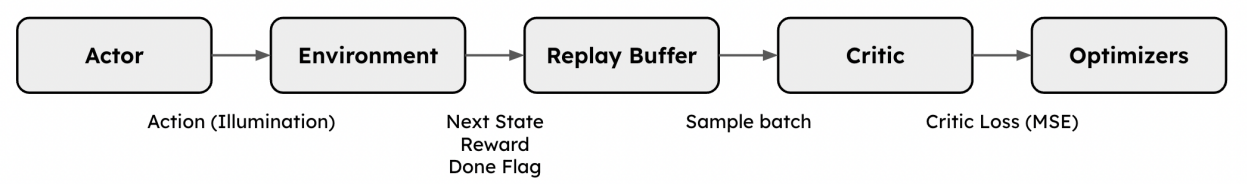


Figure 2: This diagram depicts the high-level architecture using the JaxGCRL codebase as a framework. The actor network is the policy generator—it receives the current state from the environment and outputs an illumination pattern. This action is applied to the environment, which simulates the OptoEGFR response to the illumination and returns the next state, reward, and a done flag to be stored in the replay buffer. The critic network estimates Q-values to evaluate the quality of the generated state-action pairs and guides the optimization of both networks.

The critic estimates the Q-values of the given state-action pairs, essentially providing an evaluation how well the illumination pattern worked in guiding the cells to the goal state. The optimizer serves to update the actor and critic based on the computed gradients to minimize the loss and continuously improve upon the networks. In this section, we will provide an overview of the implementation of each part of the system.

4.1. Environment

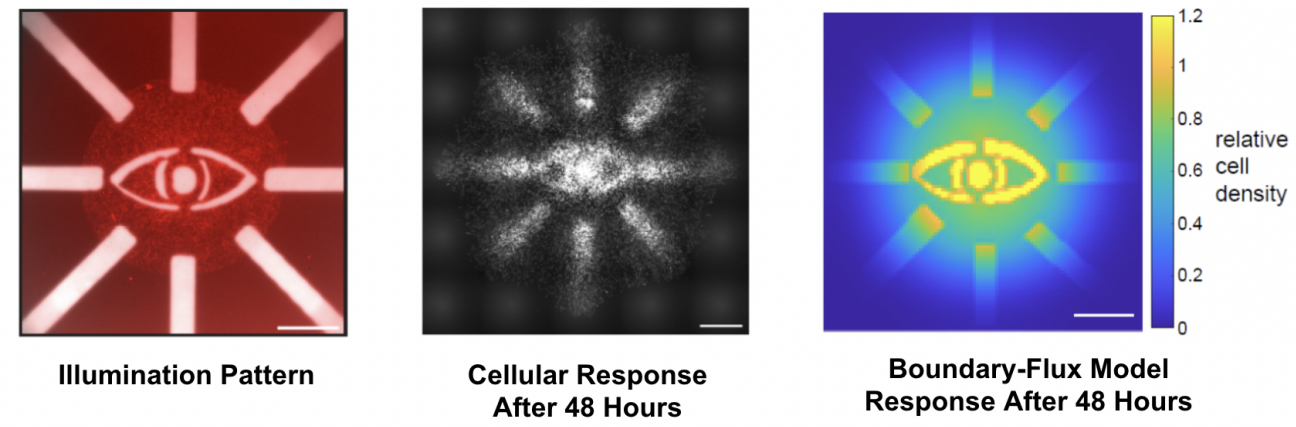


Figure 3: The leftmost figure shows the illumination pattern applied to a circular tissue. The middle figure shows imaging of cells after 48 hours of exposure to the illumination. The rightmost figure shows the boundary-flux model's prediction of the cell densities after 48 hours of exposure to the illumination [28].

Suh et al. developed a boundary-flux model of OptoEGFR behavior using MATLAB. The boundary-flux model accounts for two main forms of movement: (1) outward diffusion, captured

by the diffusion parameter D, and (2) flux at the tissue boundaries into the illuminated spaces, captured by the boundary flux parameter k. Based on experimental results, the parameters were set to be $D = 50\ \mu\text{m}^2/\text{min}$ and $k = 1\ \text{min}^{-1}$ [28]. The boundary-flux model, shown in Figure 3, served as the basis for creating the simulation environment. We obtained the MATLAB code for the boundary-flux model from Suh et al., and we will now delve into the logic underlying the ODE equations that guided our creation of the Python environment [28].

Outward diffusion occurs randomly, as cells naturally spread along concentration gradients even in the absence of external stimuli [28]. For cells at the interior of the tissue layer, the diffusion is approximated using a discrete Laplacian. First, we will denote $y_{i,j}(t)$ as the cell density at position (i, j) and time t. Then, the diffusion is given by:

$$\left.\frac{dy_{i,j}}{dt}\right|_{\text{diff}} = \frac{D}{\Delta x^2}\Big(y_{i-1,j} + y_{i+1,j} + y_{i,j-1} + y_{i,j+1} - 4y_{i,j}\Big), \tag{1}$$

When an illumination pattern is applied, the edges of the pattern form boundaries that affect cell movement [28]. We denote these boundaries by specifying the indices they come "from" and "to" as (i_{uf}, j_{uf}) and (i_{ut}, j_{ut}), respectively. For instance, upward flux changes these rates as:

$$\left.\frac{dy}{dt}\right|_{\text{up, from}}(i_{uf}, j_{uf}) = -\frac{f}{\Delta x}y(i_{uf}, j_{uf}), \tag{2}$$

$$\left.\frac{dy}{dt}\right|_{\text{up, to}}(i_{ut}, j_{ut}) = +\frac{f}{\Delta x}y(i_{uf}, j_{uf}), \tag{3}$$

where f is the flux coefficient. The same line of reasoning follows for the dynamics that occur for flux that goes down, left, and right. Now, to combine the impact of diffusion and boundary flux, the rate of change at any given position (i, j) is given by:

$$\frac{dy_{i,j}}{dt} = \left.\frac{dy_{i,j}}{dt}\right|_{\text{diff}} + \sum_{\text{directions}} \left.\frac{dy_{i,j}}{dt}\right|_{\text{flux}}. \tag{4}$$

To ensure compatibility with the JaxGCRL codebase, we developed a custom simulation environ-

ment using the ODE equations defined in the original boundary-flux model, converting the original MATLAB logic into Python and aligning the environment to be compatible with the training and networks code in JaxGCRL. The environment is defined in the **`opto_env.py`** module, and consists of a 100×100 grid, where each square on the grid is assigned a floating-point value corresponding to the cell density in that region. The illumination pattern is represented as a 4-dimensional action vector, where each dimension corresponds to the light intensity applied in a specific quadrant of the environment. At each step, we generate plots for the illumination pattern, cell densities, and target state. These are saved as image files as a visual indicator of the controller's performance. We will now outline the observation space, how the environment is initialized and reset, and how the agent interacts with the environment in each time step.

4.1.1. Observation Space

The observation space is a concatenated vector of 30,398 total dimensions. Our goal in creating the observation vector was to ensure compatability with the dimensions expected by the existing modules in the JaxGCRL codebase, such as **`train.py`**, **`networks.py`**, and **`losses.py`**. Table 1 outlines each component of the observation space.

Component	Dimensions	Description
Grid	10,000	Flattened current cell densities at a given timestep
Illumination Pattern	10,000	Flattened version of the current illumination pattern
Target Density	10,000	Target cell density distribution (goal state)
Dummy Zeros	396	Additional information needed for consistency with the dimensions expected by the existing architecture
Goal Indicators	2	Coordinates of the goal densities in the grid

Table 1: Observation Vector Components

4.1.2. Initialization

Resetting the environment entails re-initializing the cell densities and the densities of the goal states:

1. **Initial Cell Densities:** To ensure that the starting points for each episode are diverse, the environment places 7×7 patches of cells at random, non-overlapping regions.

2. **Goal States:** The cell densities for the goal states are also generated by placing 7×7 patches at random locations in the environment.
3. **Illumination Patterns:** The illumination pattern is initialized to be zero across all of the quadrants in the action space, which indicates the starting point of no illumination being applied.
4. **State Initialization:** The `State` object is returned to the agent and includes the initial cell densities, illumination pattern, the goal state, and any additional information such as the dummy zeros and goal indicators outlined in Table 1.

4.1.3. Steps

The agent interacts with the environment to iteratively improve the generated illumination pattern. We will provide a high-level overview of each steps:

1. The illumination pattern is applied to the grid, and the controller adjusts the illumination intensity in each quadrant according to the policy.
2. The illumination changes the densities of cells in corresponding regions by a small factor according to the boundary-flux dynamics determined experimentally by Suh et al. [28]. More specifically, the environment accounts for two key processes (see Section 4.1 for details):
   - **Outward Diffusion:** At any given time, cells randomly diffuse from areas of high to low density.
   - **Boundary Flux:** At the illumination borders, directed flux will alter the cell densities to simulate collective movement toward regions that are illuminated.
3. The reward function is intrinsic (see Section 4.2 for details), and encourages the controller to prioritize diversity and exploration in generating illumination patterns.
4. The environment updates the observation vector with the new state, which is then provided to the agent for the next action.

4.2. Rewards

For the unsupervised setting, we use intrinsic rewards that encourage the agent to explore and learn new representations of the environment. Rather than using traditional methods like mean squared error (MSE), the reward function encourages the agent to interact extensively with the environment and prioritize the diversity of the states visited.

At each time step t, the intrinsic reward r_t encourages the agent to explore new states. More specifically, the reward is inversely proportional to the square root of the visitation count of the current state. We discretize the continuous state s_t using a state representation function $\phi(s_t)$ [23, 12]. The reward is defined as follows:

$$r_t = \beta \cdot \frac{1}{\sqrt{n(\phi(s_t)) + 1}} \tag{5}$$

where $n(\phi(s_t))$ is the number of visits, up to a time t, to the state $\phi(s_t)$ and β is a scaling factor. This provides higher rewards when the agent explores less-visited states—the inverse relationship helps strike a balance between exploration and exploitation. To prevent the policy from becoming too deterministic, we encourage exploration using entropy regularization [10]:

$$\mathscr{L}_{\text{entropy}} = -\lambda H(\pi(\cdot|s_t)) \tag{6}$$

where λ is a hyperparameter and $H(\pi(\cdot|s_t))$ is the entropy of the distribution of actions given a state s_t. Having a higher entropy helps keep a level of randomness in the actions recommended by the policy, which helps the agent explore new strategies rather than potentially converging on ones that yield suboptimal illumination patterns. The overall training objective is defined as:

$$\mathscr{L} = E\left[\sum_{t=0}^{T} \gamma^t \left(r_t + \mathscr{L}_{\text{entropy}}\right)\right] \tag{7}$$

where γ is a discount factor.

4.3. Network Architecture

The hierarchical controller consists of a high-level planner and a low-level controller. The neural networks are all built as multi-layer perceptrions (MLPs) in the **`networks.py`** module, which is built upon the JaxGCRL codebase and include the policy network, State-Action (SA) encoder, Goal (G) encoder, and controller.

4.3.1. Policy Network

Based on the current state of the environment, the policy network generates the initial illumination patterns. The MLP takes in inputs such as cell densities and the current illumination pattern, and then outputs a parametric distribution for the actions (updated illumination patterns). The hidden layers are of size 256 and the network aims to capture non-linear relationships between the state of the input and the actions it must output. We also apply a ReLU activation function and use layer normalization for stable training.

4.3.2. SA Encoder and G Encoder

The SA Encoder and G Encoder are responsible for embedding the state-action pairs and goal states, respectively, into a lower-level, latent representation space. This allows the agent to easily compare different states and align them with the desired goal states. The architecture is similar to the policy network outlined above, and encourages the agent to focus more on actions that are likely to guide the agent toward the desired goal states [2].

4.3.3. Controller Network

The controller network is responsible for refining the illumination patterns that are generated by the policy network. The controller network takes in the encoded goal representations and then makes adjustments to the generated illumination pattern, which is crucial in order for the system to be responsive to real-time changes in cell densities. Ultimately, the controller network aims to minimize the difference between the actual cell densities and the desired goal states.

5. Evaluation

5.1. Experiment Design

The evaluation uses the mechanism outlined in the **`evaluator.py`** module derived from JaxGCRL. This uses 128 parallel evaluation environments to both speed up the evaluation process and ensure that the results we obtain are actually representative of the agent's performance over a wide variety of scenarios. In each step, the agent uses the policy to choose actions based on its current observations of the environment, which, in turn, impacts the illumination patterns generated and the cell densities. All training runs were carried out using a batch size of 256 and a discount factor of 0.99 at 10M environment steps. We also use a contrastive loss function with an L2 energy function to optimize the policy [2].

At each time step, the observation vector is comprised of the cell densities, the illumination pattern, target densities of the goal state, dummy zeros, and goal indicators, as outlined in Table 1. To train the agent in a variety of scenarios, we vary the initial conditions and goal densities across episodes. The actions taken by the agent are guided by the policy network, and this formulates the illumination pattern that will be applied to the grid.

5.2. Metrics

The JaxGCRL codebase supports the logging of various metrics to quantitatively evaluate the agent, which we outline in Table 2. We are primarily interested in the success rate, which measures if the agent reached the goal at least once during the episode [2]. We also monitor the actor and critic losses internally to ensure that the training process is guiding the agent to learn as intended, as well as the contrastive loss to evaluate the agent's performance in both prioritizing diversity and discovering effective strategies.

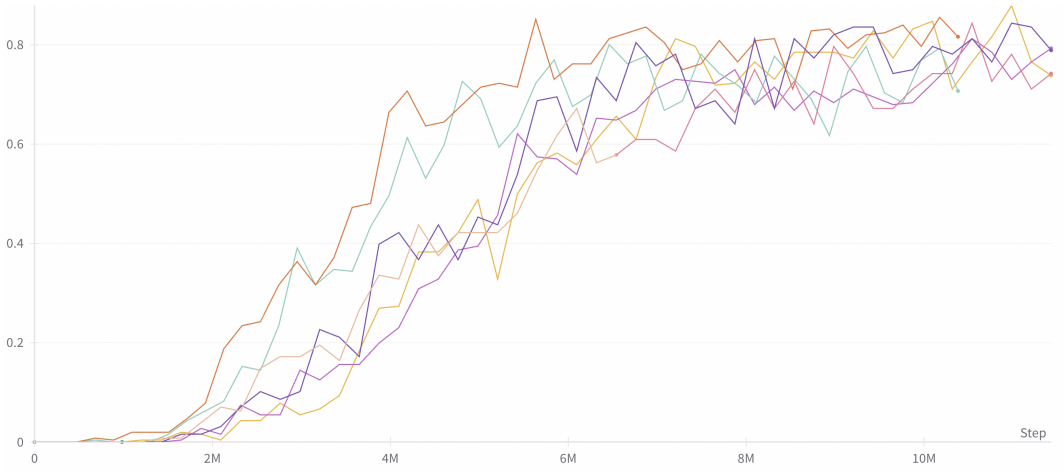
Metric	Description
Success Rate	Proportion of episodes where the agent successfully reaches the goal [2]
Episode Length	Steps to achieve sufficient exploration before the episode terminates [2]
Policy Entropy	Randomness in action selection [29]
Contrastive Loss	Alignment between the agent's internal representations and environmental states [3]
Actor and Critic Losses	Tracks training progress of the policy (actor) and value (critic) networks [29]

Table 2: Evaluation Metrics

5.3. Results

Using the logging capabilities supported by JaxGCRL, we were able to visualize key metrics on Weights & Biases. The success rate for the training setting described in Section 5.1 is shown in Figure 4. In the first graph, we show the success rate for a single run, where the highest success rate reached over the course of 10M timesteps is 81.25% and this rate follows an overall increasing trend. We also show in the second graph, the success rate for additional training runs, to show that the controller was able to achieve relatively consistent success rates across different runs. The controller is able to achieve and maintain a success rate above 80% in most runs after approximately 6M timesteps. This suggests that the controller was able to achieve a relatively high level of success in learning how to generate optimal illumination patterns to guide the OptoEGFR cells to form the desired goal states.

(a) Success Rate (Single Run)

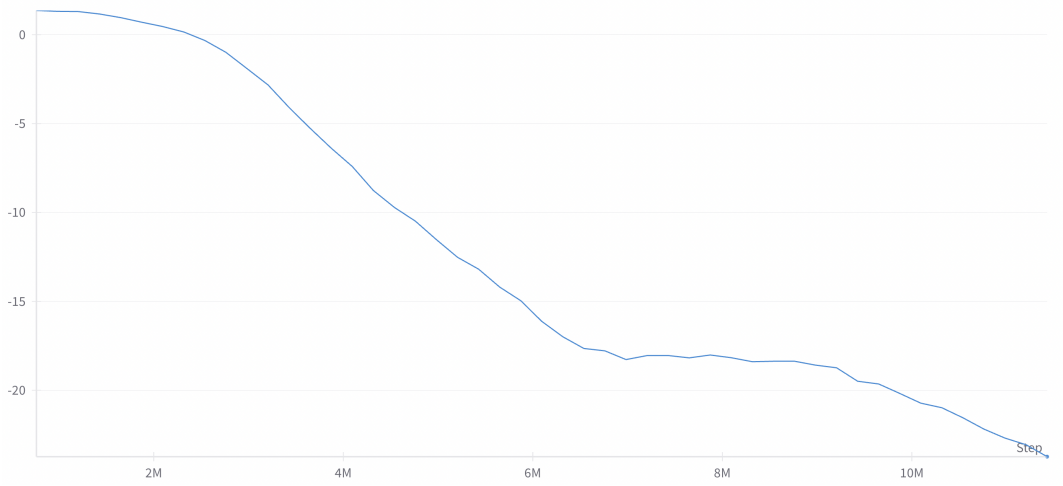


(b) Success Rate (Multiple Runs)

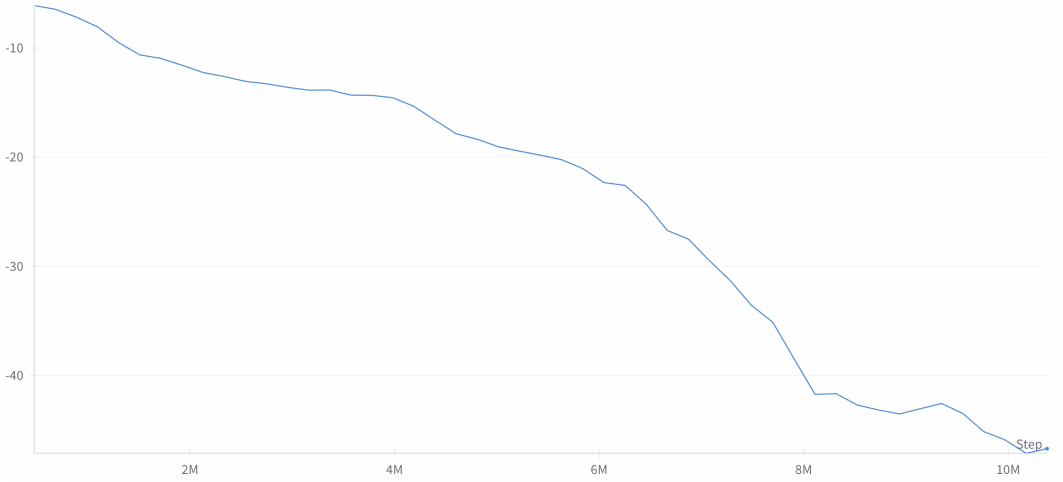
Figure 4: Success Rate

Furthermore, we analyzed the contrastive loss components: uniformity loss and alignment loss. As shown in Figure 5, both the uniformity and alignment losses follow a decreasing trend. The decreasing uniformity loss suggests that negative state-action pairs are being pushed further apart, which indicates that the controller is capable of generating a variety of illumination patterns instead of just converging to a small set of similar patterns. The alignment loss measures how closely the illumination patterns generated by the controller correlate with the desired goal states [32]. The decreasing trend suggests that positive pairs are being pushed closer together, which indicates that the controller's ability to generate effective and efficient illumination patterns is improving throughout the training process. Overall, the decreasing trend we observe in both contrastive loss

components suggests that even in the absence of a pre-defined reward function, the controller is able to simultaneously explore a diverse set of strategies and focus on strategies that are able to generate the most optimal illumination patterns to achieve the desired goal states.



(a) Uniformity Loss



(b) Alignment Loss

Figure 5: Contrastive Losses

We show an example of the controller's performance in Figure 6, where each panel is a 100×100 grid. The top left image represents the initial densities of the cells, where most of the cells are clustered in the middle of the grid in a rectangular shape, and the cell densities are lower in the surrounding regions. The top right image is a pre-determined goal state, which is a circular shape with a higher cell density in the middle that ebbs out at the outer bounds of the shape. Finally, the

bottom image is the illumination pattern output by the controller to guide the cells from the initial state to the goal state. Because the cells had a higher initial density in the region corresponding to the upper half of the circle, the illumination was more intense in the lower half of the pattern in order to effectively guide cells from more sparse regions.

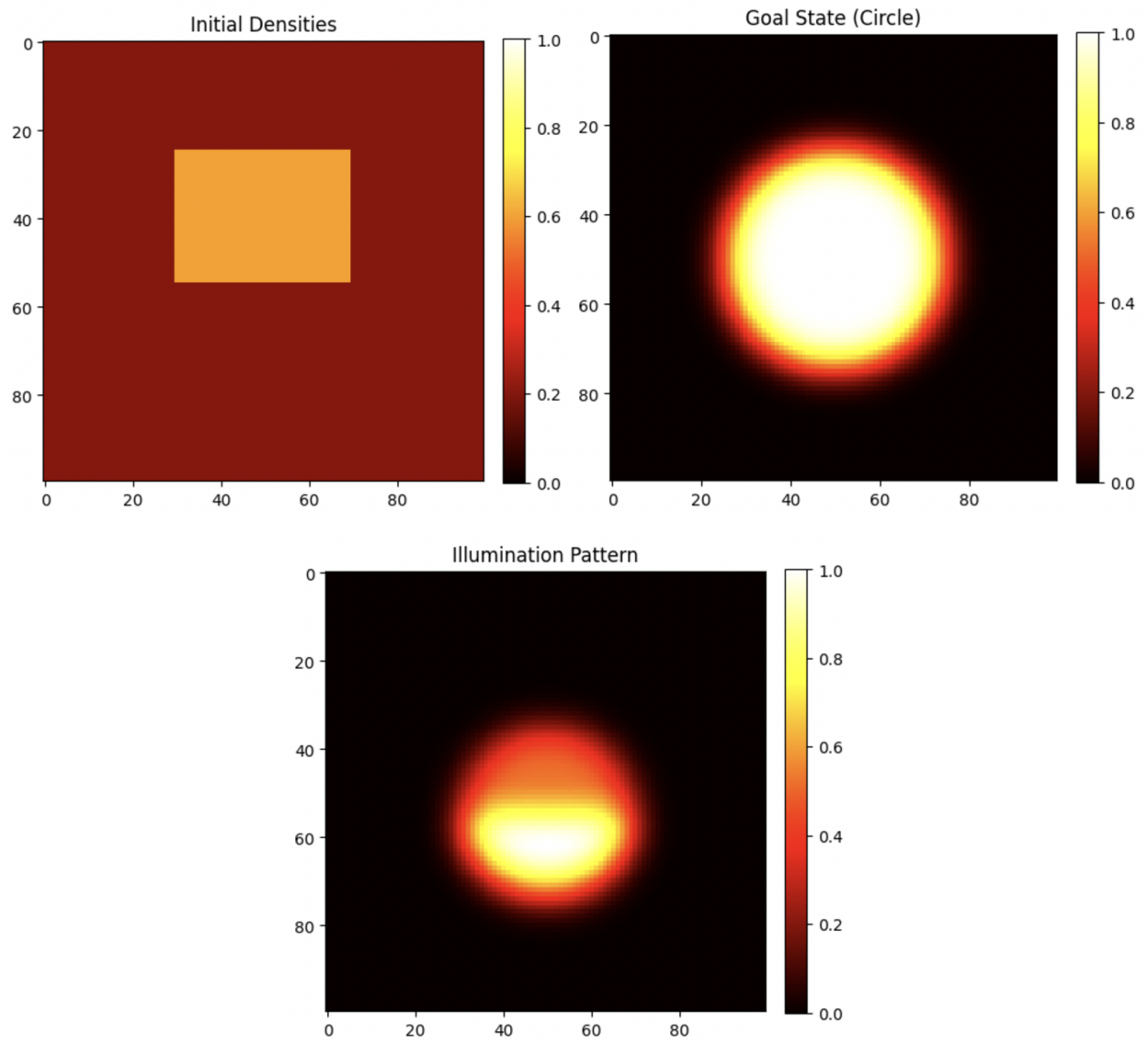


Figure 6: This illustrates a sample rendering generated by the controller. The top left image is the initial cell density, the top right image is the goal state for cell densities (a circular distribution), and the bottom image is the optimal illumination pattern determined by the controller. The plots are 100×100, and each square on the grid is colored according to the corresponding cell density.

Figure 7 shows a few illumination patterns that the controller experimented with to achieve the same goal state depicted in Figure 6. The top two images show illumination patterns where the controller explored using different illumination intensities and spatial distributions to guide the cells to the goal state. The pattern in the bottom image demonstrates the controller's ability to use a diverse set of strategies to guide cell movement—it simultaneously adjusts the spatial distribution and the illumination intensities of the pattern in an attempt to optimize cell movement. The intrinsic reward function outlined in Section 4.2 encourages the controller to explore a diverse set of strategies, which is demonstrated in the diversity of the patterns generated.

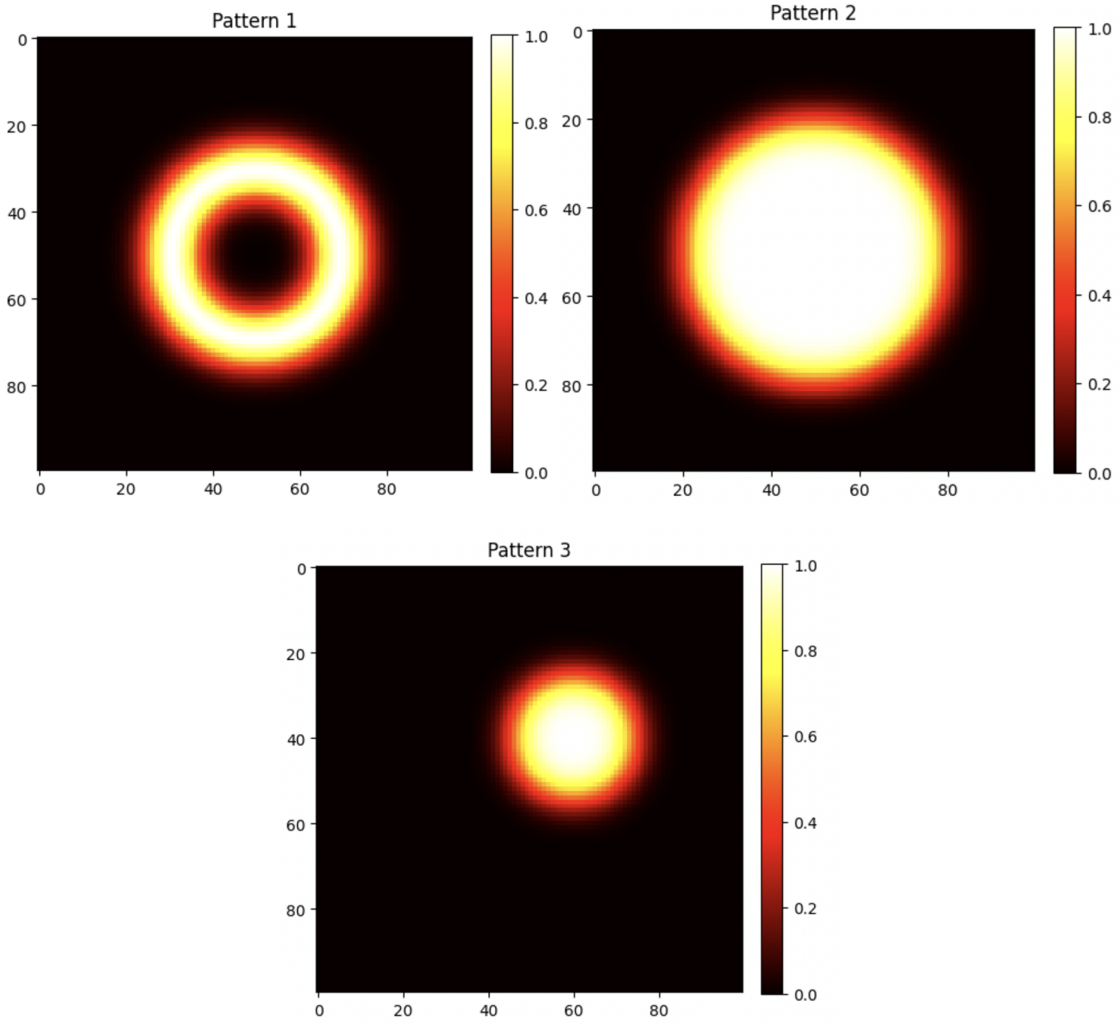


Figure 7: This image depicts a few alternative illumination patterns that the controller tested out for the initial densities and goal state outlined in Figure 6. The plots are 100×100, and each square on the grid is colored according to the corresponding cell density.

6. Summary

6.1. Conclusions

In this paper, we present a hierarchical unsupervised RL controller that leverages the JaxGCRL framework to dynamically generate optimal illumination patterns that are able to guide cells to collectively migrate to a goal conformation. We show that the controller is able to achieve a relatively high success rate while minimizing their contrastive losses, which demonstrates that over the course of the training process, the controller is able to explore a diverse set of strategies and apply optimal strategies in generating the final illumination patterns. This has key implications in bridging the gap between using OptoEGFR to control cell movement and applying this innovative tool to real-life, high-stakes use cases like wound regeneration.

6.2. Limitations and Future Work

While the controller was able to achieve a high rate of success in generating optimal illumination patterns, there are still several limitations and areas for improvement that can guide future work. First, the environment that the controller is built on is based on the boundary-flux model developed by Suh et al., which was developed based on experimental data. However, this model serves as a simplified simulation that does not account for more complex tissue features like fluidization or jamming that arises from areas of high density [28]. Future work would focus on developing a model that is able to capture these complexities and serve as a better basis for modeling cell migration dynamics. In addition, while the boundary-flux model parameters were determined experimentally, this controller is heavily reliant on simulated data from this simplified model. With further experimentation, we could further calibrate the model to improve the reliability of the controller. Even though the controller is able to achieve a relatively high success rate, high-stakes applications like wound regeneration would necessitate a higher degree of precision.

Furthermore, the current environment is modeled as a 100×100 grid, which is an oversimplifica-

tion of the tissue scale. In future work, we could focus on expanding this grid size to capture more complex and extensive environments, which would open up possibilities in creating more detailed illumination patterns. Future work could also focus on more intentional environment initialization to capture conditions that more accurately reflect the realities in biological settings. Currently, the initial densities are set in a randomized manner, which does not fully capture the nature of cell densities that are found in real biological systems.

6.3. Acknowledgements

I would like to acknowledge Professor Benjamin Eysenbach for his incredible support, enthusiasm, and constant willingness to help me and my peers. I would also like to acknowledge Professor Jared Toettcher and Richard Thornton for their insights on OptoEGFR that guided me throughout my IW.

References

[1] J. Bishop *et al.*, "Proto-oncogenes and plasticity in cell signaling," *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 59, pp. 165–171, 1994.

[2] M. Bortkiewicz *et al.*, "Accelerating goal-conditioned rl algorithms and research," *arXiv preprint arXiv: 2408.11052*, 2024.

[3] T. Chen *et al.*, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[4] F. H. C. Crick, "Thinking about the brain," *Scientific American*, vol. 241, no. 3, pp. 219–233, 1979. Available: http://www.jstor.org/stable/24965297

[5] K. Deisseroth, "Optogenetics," *Nature Methods*, vol. 8, no. 1, pp. 26–29, 2011.

[6] D. Del Vecchio, A. J. Dy, and Y. Qian, "Control theory for synthetic biology: Recent advances in system characterization, control design, and controller implementation for synthetic biology," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 32–62, 2018. Available: https://murray.cds.caltech.edu/Control_Theory_for_Synthetic_Biology%3A_Recent_Advances_in_System_Characterization%2C_Control_Design%2C_and_Controller_Implementation_for_Synthetic_Biology

[7] B. Eysenbach *et al.*, "Diversity is all you need: Learning skills without a reward function," in *International Conference on Learning Representations*, 2019. Available: https://openreview.net/forum?id=SJx63jRqFm

[8] L. Fenno, O. Yizhar, and K. Deisseroth, "The development and application of optogenetics," *Annual Review of Neuroscience*, vol. 34, pp. 389–412, 2011.

[9] D. Ghosh, A. Gupta, and S. Levine, "Learning actionable representations with goal-conditioned policies," in *International Conference on Learning Representations*, 2019. Available: https://arxiv.org/abs/1811.07819

[10] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[11] S. He *et al.*, "Wasserstein unsupervised reinforcement learning," 2021. Available: https://arxiv.org/abs/2110.07940

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[13] H. Hou *et al.*, "Using deep reinforcement learning to speed up collective cell migration," *BMC Bioinformatics*, vol. 20, no. Suppl 18, p. 571, 2019.

[14] J. LaChance *et al.*, "Learning the rules of collective cell migration using deep attention networks," *PLOS Computational Biology*, vol. 18, no. 4, p. e1009293, 2022. Available: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009293

[15] M. Laskin *et al.*, "Urlb: Unsupervised reinforcement learning benchmark," *arXiv preprint arXiv:2110.15191*, 2021. Available: https://arxiv.org/abs/2110.15191

[16] B. Lin *et al.*, "Synthetic spatially graded rac activation drives cell polarization and movement," *Proceedings of the National Academy of Sciences*, vol. 109, no. 52, pp. E3591–E3600, 2012.

[17] M. Lintz, A. Munoz, and C. A. Reinhart-King, "The mechanics of single-cell and collective migration of tumor cells," *Journal of Biomechanical Engineering*, vol. 139, 2017.

[18] C. Matellan and A. E. del Río Hernández, "Engineering the cellular mechanical microenvironment – from bulk mechanics to the nanoscale," *Journal of Cell Science*, vol. 132, no. 9, p. jcs229013, 2019.

[19] S. Meloche and J. Pouysségur, "The erk1/2 mitogen-activated protein kinase pathway as a master regulator of the g1- to s-phase transition," *Oncogene*, vol. 26, pp. 3227–3239, 2007.

[20] D. J. Montell, "Morphogenetic cell movements: diversity from modular mechanical properties," *Science*, vol. 322, no. 5907, pp. 1502–1505, 2008.

[21] A. A. Pevtsov and A. V. Kolesnikov, "Control of dynamic systems under input and output constraints," *Automation and Remote Control*, vol. 84, p. 451–463, 2023. Available: https://link.springer.com/article/10.1134/S0005117923040069

[22] I. Queinnec, S. Tarbouriech, and G. Garcia, "Analysis and control of dynamical biological systems in presence of limitations," in *Biology and Control Theory: Current Challenges*. Springer, 2007, pp. 317–338. Available: https://link.springer.com/chapter/10.1007/978-3-540-71988-5_13

[23] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2007, pp. 1177–1184.

[24] A. J. Ridley *et al.*, "Cell migration: Integrating signals from front to back," *Science*, vol. 302, no. 5651, pp. 1704–1709, 2003.

[25] M. T. Rosenstein and A. G. Barto, "Supervised actor-critic reinforcement learning," in *Handbook of Learning and Approximate Dynamic Programming*, J. Si *et al.*, Eds. Piscataway, NJ: IEEE Press, 2004, pp. 359–380. Available: https://ieeexplore.ieee.org/document/5273614

[26] Y. Shin *et al.*, "Spatiotemporal control of intracellular phase transitions using light-activated optodroplets," *Cell*, vol. 168, no. 1-2, pp. 159–171.e14, 2017.

[27] T. P. Soltoff *et al.*, "Erbb3 is involved in activation of phosphatidylinositol 3-kinase by epidermal growth factor," *Molecular and Cellular Biology*, vol. 14, pp. 3550–3558, 1994. Available: https://doi.org/10.1128/mcb.14.6.3550-3558.1994

[28] K. Suh *et al.*, "Large-scale control over collective cell migration using light-controlled epidermal growth factor receptors," *bioRxiv*, 2024. Available: https://www.biorxiv.org/content/early/2024/05/31/2024.05.30.596676

[29] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[30] J. E. Toettcher *et al.*, "Light-based feedback for controlling intracellular signaling dynamics," *Nature Methods*, vol. 10, no. 4, pp. 353–358, 2013.

[31] J. E. Toettcher *et al.*, "Using optogenetics to interrogate the dynamic control of signal transmission by the ras/erk module," *Cell*, vol. 155, no. 6, pp. 1422–1434, 2013. Available: https://doi.org/10.1016/j.cell.2013.11.004

[32] T. Wang and A. A. Efros, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020, pp. 873–882.

[33] S. Werner and R. Grose, "Regulation of wound healing by growth factors and cytokines," *Physiological Reviews*, vol. 83, no. 3, pp. 835–870, 2003.

[34] A. E. Wolf *et al.*, "Short-term bioelectric stimulation of collective cell migration in tissues reprograms long-term supracellular dynamics," *PNAS Nexus*, vol. 1, no. 1, p. pgac002, 2022.

[35] Y. Wu *et al.*, "A genetically encoded photoactivatable rac controls the motility of living cells," *Nature*, vol. 461, no. 7260, pp. 104–108, 2009.

[36] T. J. Zajdel *et al.*, "SCHEEPDOG: Programming Electric cues to Dynamically Herd Large-Scale Cell Migration," *Cell Systems*, vol. 10, no. 6, pp. 506–514.e3, 2020.

[37] Y. Zhang *et al.*, "A deep reinforcement learning model based on deterministic policy gradient for collective neural crest cell migration," 2020.