IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 August 2022, accepted 20 September 2022, date of publication 26 September 2022,
date of current version 30 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3209260

RESEARCH ARTICLE

Flow-Based Reinforcement Learning

DILINI SAMARASINGHE, MICHAEL BARLOW, AND ERANDI LAKSHIKA
School of Engineering and IT, University of New South Wales, Canberra, ACT 2612, Australia
Corresponding author: Dilini Samarasinghe (d.samarasinghe@adfa.edu.au)

ABSTRACT This paper presents a novel Flow-based reinforcement learning strategy to model agent systems that can adapt to complex and dynamic problem environments by incrementally mastering their skills. It is inspired by the psychological notion of Flow that describes the optimal mental state experienced by an individual when they are fully immersed in a task and find it intrinsically rewarding to engage with. The proposed model presents an algorithm to describe the Flow experience such that agents can be trained through finer distinctions to the challenges across training time to maintain them in the Flow zone. In contrast to the traditional and incremental learning approaches that suffer from limitations associated with overfitting, the Flow-based model drives agent behaviours not simply through external goals but also through intrinsic curiosity to improve their skills and thus the performance levels. Experimental evaluations are conducted across two simulation environments on a maze navigation task and a reward collection task with comparisons against a generic reinforcement learning model and an incremental reinforcement learning model. The results reveal that these two models are prone to overfit under different design decisions and loose the ability to perform in dynamic variations of the tasks in varying degrees. Conversely, the proposed Flow-based model is capable of achieving near optimal solutions with random environmental factors, appropriately utilising the previously learned knowledge to identify robust solutions to complex problems.

INDEX TERMS Flow, reinforcement learning, incremental learning, machine learning, artificial intelligence.

I. INTRODUCTION

Reinforcement learning (RL) is a prominent artificial intelligence (AI) technique that has been used in modeling agent behaviour in complex environments. RL models have been exploited in diverse agent-based systems that tackle problems such as coordinated exploration [1], [2], path planning [3], [4], collision avoidance [5], locomotion control [6], and other complex decision making tasks [7], [8], [9], [10] with both virtual and physical applications. However, a known limitation of the existing RL-based agent models is the difficulty in adapting to dynamic and uncertain conditions. This is primarily caused by the increased complexity of operation associated with changes in the environment [11], [12], [13].

This paper investigates a novel Flow-based RL strategy which allows agent systems to adapt to complex environments by incrementally mastering their skills. In psychology, *Flow* refers to the mental state experienced by an individual when they are fully immersed in a task and find it intrinsically rewarding to engage with. While there has always been an awareness among people of the feeling of immersion, loss of self-consciousness, and happiness experienced while being fully engaged in a task they like, the concept was first coined by the psychologist Mihaly Csikszentmihalyi [14]. The key dimensions of any experience of a task are the challenges the task brings, and skills required to achieve them. One deviates from a Flow state of mind when they feel: anxious, due to a challenge being beyond their reach; or bored, due to a challenge being easily achievable compared to their current skill level. If the challenges and the skill levels increase proportionally within the Flow zone, it can facilitate a sense of discovery driving one with an intrinsic motivation for higher performance levels.

We adapt this concept of Flow in training artificial agents within a reinforcement learning model by making finer distinctions to the challenges across training time to maintain agents in a Flow zone. It can overcome intrinsic challenges such as overfitting and catastrophic forgetting associated with

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

adaptation to dynamic environments [15] by driving agents through both external goals as well as internal curiosity to explore novel solutions. We demonstrate that an agent trained with a Flow-based strategy is more robust than one trained with a traditional or incremental reinforcement strategy and can better perform in any random variation of the task environment they may encounter in future. This has major implications for future in developing resilient intelligent agents and simulation technologies for modelling decision making and control strategies in diverse application environments. The contributions of this paper in this regard are as follows:

- A novel Flow-based RL algorithm is proposed to enhance the learning ability of artificial agents in complex and dynamic environments.
- A measure of identifying the Flow zone is introduced using novelty of the solution identified.
- Simulations are conducted in two environments focused on maze navigation, and reward collection.
- Evaluations are presented with comparisons against a traditional and an incremental RL model with dynamic tasks to investigate the proposed model.

The rest of the paper is organised as follows. Section II summarises the relevant existing literature. Flow as described by Csikszentmihalyi and the proposed adaptations of the notion in the AI domain with the framework for Flow-based RL model are presented in Section III. The experimental setups and evaluations are illustrated in Section IV. Finally, Section V concludes the paper with a discussion of the results and possible future directions.

II. RELATED WORK

Designing artificial agents that can adapt to dynamic and complex problem environments has often been discussed as a critical challenge to be addressed in the AI domain for decades [16], [17], [18], [19], [20]. It has implications in formalising a vast array of real-life applications from domestic ground robots to SAR (search and rescue) drones and self-driving vehicles. Researchers have explored several approaches to overcome the challenge of developing robust agent models by approaching complex problems through solving simpler versions. Incremental learning [21] has emerged as a potential solution where controller behaviours are learned by progressively increasing the scope or the complexity of the task. It has been used to refine the actions of an agent incrementally over episodes such that a suitable policy can be synthesised for achieving the ultimate complex goal [22]. Using a gradient descent to incrementally increasing the number of agents involved and complexity of the task [23]; generating RL detector agents to detect environmental changes and update the value functions and thus the previous policy to suit the new environment [24]; and lifelong incremental learning through a library of an infinite mixture of parameterised environment models [25] are some approaches where incremental learning with RL has been used for agent modelling. In a similar vein, transfer learning [26] is an approach that uses knowledge gained in a previous task to subsequently address a related but different task. It has been adapted with evolutionary transfer RL frameworks [27]; policy intersection to allow an external policy influence the RL agent [28]; and with fine-tuning where tasks are parameterised by their reward functions [29] among other applications. Self-learning adaptive dynamic programming [30] is also experimented in this regard as a means of eliminating the explicit external reward scheme by encouraging agents to learn internal rewards dynamically based on the problem presented. The use of abstractions or modular RL is another approach to solve complex problems through tasks being subdivided into multiple simpler modules to be learned independently and combined [31], [32], [33], [34].

The key limitation with these existing approaches is that they are primarily goal oriented. The agent behaviour is directed towards achieving a dynamic goal through refinement of action, and little thought is given to the learning process in terms of balancing skills and challenges. As a result, they are not capable of building a general awareness of the environment that can later be utilised under changed conditions; rather they tend to overfit to or forget the accumulated knowledge [15], [35] which leads to deterioration of performance as the model is presented with more complex challenges. Such a model is incapable of developing a broad awareness of the environment that they are performing in, which can make it prone to failure when the environment changes despite being good at achieving dynamic goals.

Flow is a notion that is not focused on external goals. An agent in Flow enjoys an optimal experience where they are intrinsically motivated towards exploring the environment and building an awareness of the task, which extends beyond a simple goal oriented mind. The concept has often been adopted in human development and education as a way to understand the conditions that make the process of learning more enjoyable and efficient from a psychological point of view [36]. It has been identified that Flow can facilitate creativity and self-actualisation in the domain of learning and problem solving for humans [37]. In the technological domains, Flow has primarily been investigated with games and gamification. The interactions between a player and the game and the operative description of game-play has been characterised in the literature through the aspects of Flow on learning and enjoyment [38], [39]. However, Flow has not received attention in the domains of agent systems and AI models. It has characteristics to be explored as a potential alternative to overcome the learning issues in dynamic environments. Being in the Flow zone indicates that an agent will not completely be goal oriented but will enjoy the experience until it can no longer attain an optimal experience through novel solutions [14]. Therefore, it can lead to artificial agents that can identify more robust and generalisable solutions to problems than too narrow and specific solutions. Therefore, the work proposed here explores how the psychological theory can be adapted in the field of artificial agents to enhance

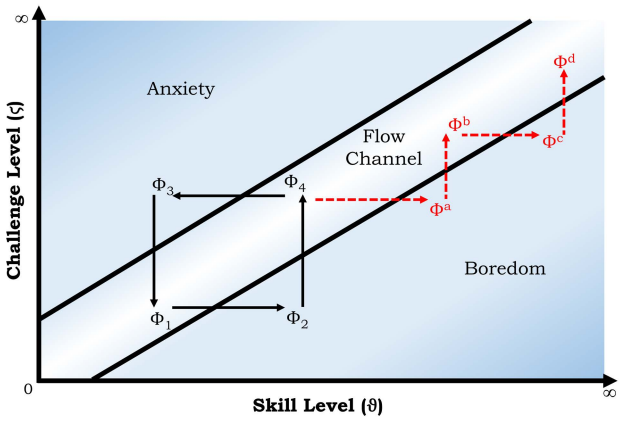


FIGURE 1. Complexity of consciousness increasing as a result of the Flow experience [14]. An experience falls out of the Flow zone if the skill levels improve without the challenge getting complex (Φ_2); or if the challenge gets increasingly complex without an opportunity to improve skills (Φ_3). In red is shown how we utilise the notion in agent systems to improve skills of agents across increasing challenges. The agent is given the opportunity to improve skills with a certain challenge level until it reaches a level of boredom (Φ^a) when the challenge is then made complex (Φ^b) bringing the agent back into the Flow zone.

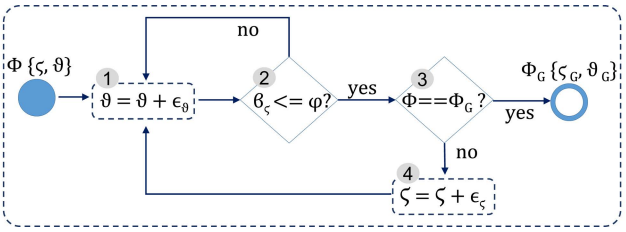


FIGURE 2. Proposed Flow-based RL model. The task Φ commences with an initial challenge level ς and a skill level ϑ. The agent keeps improving its skill level by an increment of ϵ_ϑ (step 1) until the boredom value β at the challenge level ς exceeds the boredom threshold φ (step 2). If the boredom threshold has been exceeded and the experience level Φ is the expected ultimate level of the system Φ_G (step 3), then the system completes the learning process. If not, the system increases the complexity of the challenge by an increment of ϵ_ς (step 4) and moves back to the learning step.

their horizon of performance. The tasks designed for evaluations in this work closely follow the requirements for Flow and investigations are conducted to derive insights into the applicability of the theory in practice.

III. FLOW-BASED REINFORCEMENT LEARNING

This section introduces the concept of Flow as discussed from a psychological perspective, and how the notion is adapted for the AI domain. The architecture for the proposed Flow-based RL model is discussed in detail along with the algorithms proposed.

A. FLOW

Flow or *optimal experience* is characterised by Mihaly Csikszentmihalyi as a sense that one's skills and the challenges at hand are felt to be in balance in an action system that is goal-oriented and rule-bound, where clear clues are provided for how well one is performing [14]. Flow activities provide a sense of growth and discovery leading to higher levels of performance and states of consciousness.

Figure 1 illustrates how the Flow experience improves the performance and pushes an individual towards more complex skill levels. The two axes in the diagram are the primary dimensions of any experience: the skill levels, and challenge levels. Considering an experience A; when the experience is first started at Φ_1, the individual will find it interesting and be in the Flow zone but both the challenge and the skill level are insignificant. If the challenge doesn't improve, the individual will eventually improve their skills over time and start getting bored (Φ_2) and fall out of the Flow zone. If they are to regain a positive experience from this task, the challenge has to be improved (Φ_4). On the other hand, if the challenge increases without enough time for an improvement in the skill level (Φ_3), it is possible for an individual to feel anxious of their poor performance, thus degrading the quality of the experience. The skills should improve for the individual to enjoy the activity again (Φ_4).

While both Φ_1 and Φ_4 are in the Flow zone, they are different from each other in terms of complexity. Φ_4 is more complex as the demand is for greater skills to address more difficult challenges. In order for an individual to remain in the Flow zone, both skills and challenges should be in constant evolution towards higher complexity.

B. PROPOSED FLOW-BASED RL MODEL

With the understanding of the concept of Flow in psychological experiences, the notion was adapted in our work to improve the learning ability of artificial agents in complex simulated environments. The goal is to maintain the agent(s) in the Flow zone continuously, such that both the challenges and their skills improve simultaneously over time until the expected level of performance for the expected level of challenge is reached. The experiences highlighted in red in Figure 1 illustrate this process. When the agent starts improving their skills for a given challenge (ς) and passes the threshold for boredom at Φ^a, the challenge level is incremented such that their experience will be at Φ^b. The agent then starts improving the skills again for the challenge to attain a higher performance until it cannot further improve and gets bored after some time (Φ^c), and the challenge is incremented again to bring the agent back into the Flow zone (Φ^d). This process is repeated until the ultimate challenge level is reached.

The proposed Flow-based RL model designed based on the said approach is illustrated in Figure 2. The task is commenced with an initial challenge level ς and a skill level ϑ. As the first step, the agent improves its skill level by an increment of ϵ_ϑ through the reinforcement learner. At the next step, the algorithm calculates a boredom value β at the challenge level ς and checks if it has exceeded the boredom threshold φ. If it has not, it suggests that the agent can still improve its performance and therefore moves

back to the learning step (step 1). If the boredom threshold is reached, the agent has reached its maximum performance level for the particular challenge and has moved out of the Flow zone and is not enjoying an optimal learning experience anymore. As the next step (step 3), the algorithm checks if the experience Φ being enjoyed by the agent at this level was the expected ultimate experience level of the system (Φ_G). If it is, then the system has completed the learning process and the agent is now capable of performing at the highest expected challenge level with the best possible level of skills ($\Phi_G\{\varsigma_G, \vartheta_G\}$). However, if it has not reached the ultimate challenge level, then the model increases the complexity of the challenge by an increment of ϵ_ς (step 4) and moves back to the learning step.

As discussed above, the model requires a method to quantify boredom and incorporate that with the reinforcement learner. Algorithm 1 details the proposed algorithm with the boredom calculation method illustrated in Algorithm 2.

The modified Flow-based RL algorithm 1 starts by initialising the Q-table (line 2) and the state, action pairs of the solution for the current challenge level ς as null (line 4). A decaying epsilon-greedy Q-learning approach [40] is used to balance the exploration versus exploitation tradeoff with action selection. For the initial rounds of learning, a relatively higher probability is assigned for selecting a random action, and as the learning improves, this exploration probability is reduced giving more chance for exploitation of the most suitable actions (lines 8-12). Every state,action pair ((s, a)) of the solution for each challenge level is recorded (line 14). At the end of identifying a solution during every episode of the challenge level, a boredom value is calculated based on all state, action pairs visited by the solution (lines 19-21). The value is then compared against a set threshold to determine if the agent has moved out of the Flow zone, and if it has, then the task environment is updated with a higher challenge level and the learning process is started from the beginning. If not, the agent still has the capacity to improve its performance, and the learning process is moved to the next episode in the same challenge level (lines 22-25). The process terminates if

Algorithm 1 Flow-Based Reinforcement Learning

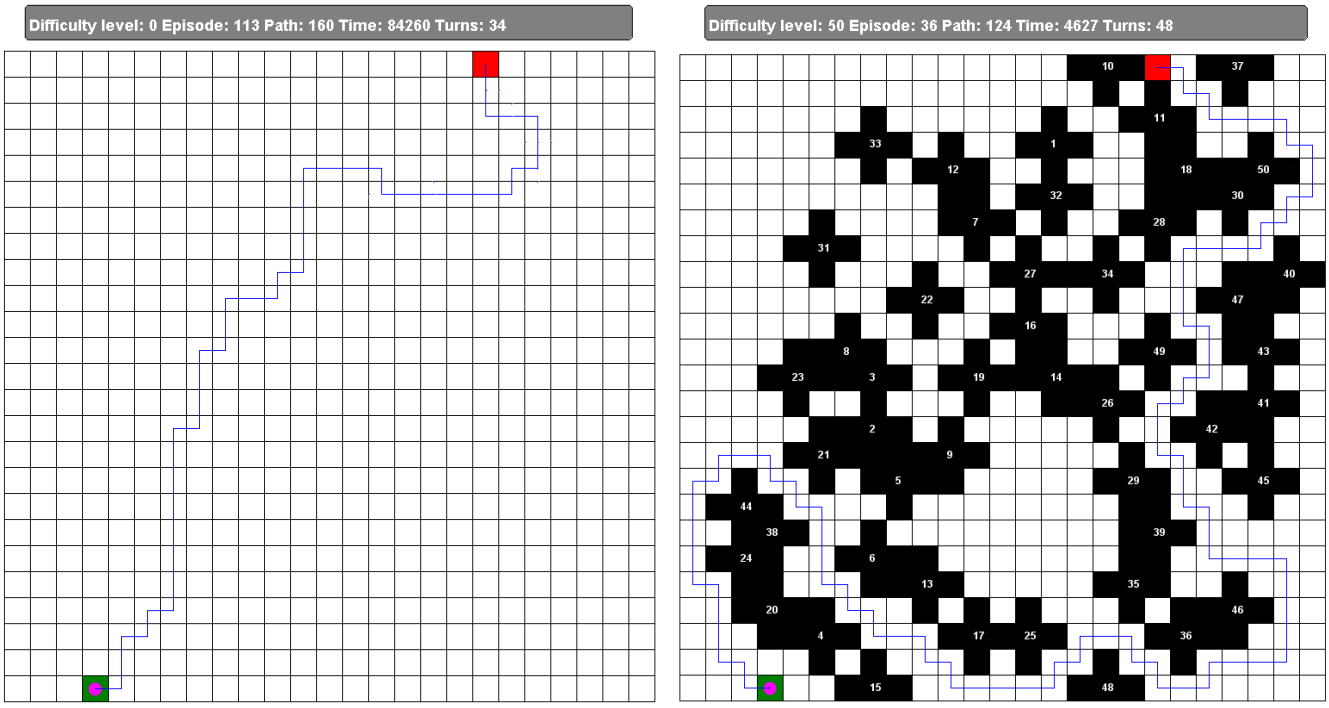
Require: ς_G : Ultimate challenge
ς : Current challenge level
ϵ_ς : Challenge increment
φ : Boredom threshold
$\rho_\varsigma(S', A')$: The state action pairs in the solutions derived for challenge ς
α : Learning rate
γ : Discount factor
ε : Decay constant
$Q(S, A)$: Q table for all state action pairs
R : Reward for each state
s : Current state
s' : New state
Ensure: β_ς : Boredom at challenge level ς

1: **procedure** FLOW-BASED RL
2: INITIALISE Q(S, A)
3: **do**
4: $\rho_\varsigma \leftarrow null$
5: **for** EACH EPISODE T **do**
6: INITIALISE STATE s
7: **do**
8: $\tau \leftarrow$ RND(0,1)
9: **if** $\tau < \varepsilon$ **then**
10: $\alpha \leftarrow$ RANDOM ACTION FROM A
11: **else**
12: $\alpha \leftarrow$ MAX Q(s)
13: $Q(s, a) \leftarrow Q(s, a) + \alpha[R + \gamma$ max Q(s', A) - Q(s, a)]
14: ADD (s, a) TO ρ_ς
15: $s \leftarrow s'$
16: **while** s is not terminal
17: $\varepsilon \leftarrow$ UPDATE(ε)
18: $b_t \leftarrow 0$
19: **for** all (s,a) pairs in ρ_ς **do**
20: $b_t \leftarrow b_t +$ BREDOM CALCULATION $(\varsigma, (s, a))$
21: $\beta_\varsigma \leftarrow b_t$ / total pairs in ρ_ς
22: **if** $\beta_\varsigma > \varphi$ **then**
23: $\varsigma \leftarrow \varsigma + \epsilon_\varsigma$
24: $\varepsilon \leftarrow$ UPDATE(ε)
25: *break*
26: **while** $\varsigma < \varsigma_G$

Algorithm 2 Boredom Calculation

Require: λ_1 : Decay constant 1
λ_2 : Decay constant 2
t : Current episode
$\sigma_\varsigma((S', A'), c)$: The number of visits (c) to each state action pair in the solutions derived for challenge ς
$\eta_{(t)}(s)$: Total number of visits to state (s) up to episode t
$\eta_{(t)}(s, a)$: Total number of visits to the state action pair (s,a) up to episode t
Ensure: $b_t(s, a)$: Boredom at episode t for state action pair (s,a)

1: **procedure** BOREDOM CALCULATION(ς, (s, a))
2: **if** (s, a) is in σ_ς **then**
3: UPDATE COUNT(σ_ς, (s, a), $(c + 1)$)
4: **else**
5: ADD (s, a) to σ_ς
6: UPDATE COUNT(σ_ς, (s, a), 1)
7: $\eta_t(s) \leftarrow$ EXTRACT VISITS (σ_ς, s)
8: $\eta_t(s, a) \leftarrow$ EXTRACT VISITS (σ_ς, (s, a))
9: $b_t(s, a) \leftarrow (\lambda_1^{\eta_t(s)} + \lambda_2^{\eta_t(s,a)})/2 - 1$
10: **return** $b_t(s, a)$



(a) Challenge Level 0 (b) Challenge Level 50

FIGURE 3. Maze navigation task. The agent is expected to navigate through the grid environment and find the shortest path from the start position (red) to the end position (green) while avoiding obstacles (black). Each challenge level introduces a new obstacle obstructing the previously identified solution, thus increasing the complexity of the challenge upto a total of 50 obstacles.

the boredom threshold has reached and the challenge level has also reached the ultimate level (line 26).[1]

The boredom calculation is done as described in Algorithm 2. The novelty calculation function in equation 1 [41] is adopted to calculated the degree of boredom experienced.

$$b_t(s, a) = (\lambda_1^{n_t(s)} + \lambda_2^{n_t(s,a)})/2 - 1 \tag{1}$$

This formula determines the frequency of visiting the state, action pair $((s, a))$ over multiple episodes of runs. At the end of every episode, the algorithm updates the number of visits to each state, action pair in the solution (lines 2-6) and then calculates the boredom for each pair (line 9). The more a certain pair is visited over multiple episodes, the less novel the solution becomes and therefore increasing the boredom value for that pair. $0 < \lambda_1, \lambda_2 < 1$ are decay constants that determine the descent rate of novelty. The average boredom of all pairs is used in the Algorithm 1 to compare with the boredom threshold. The boredom values range between [-1,0], 0 being a solution completely novel and -1 being a solution that is not novel at all. The boredom threshold is set to -1. Therefore, the solutions start with a value of 0 at the first episode of each challenge and once they reach a value of -1 with no novelty in the state,action pairs selected, the challenge is incremented to the next level.

[1]codebase available in https://doi.org/10.24433/CO.4345952.v1

IV. EXPERIMENTAL EVALUATIONS

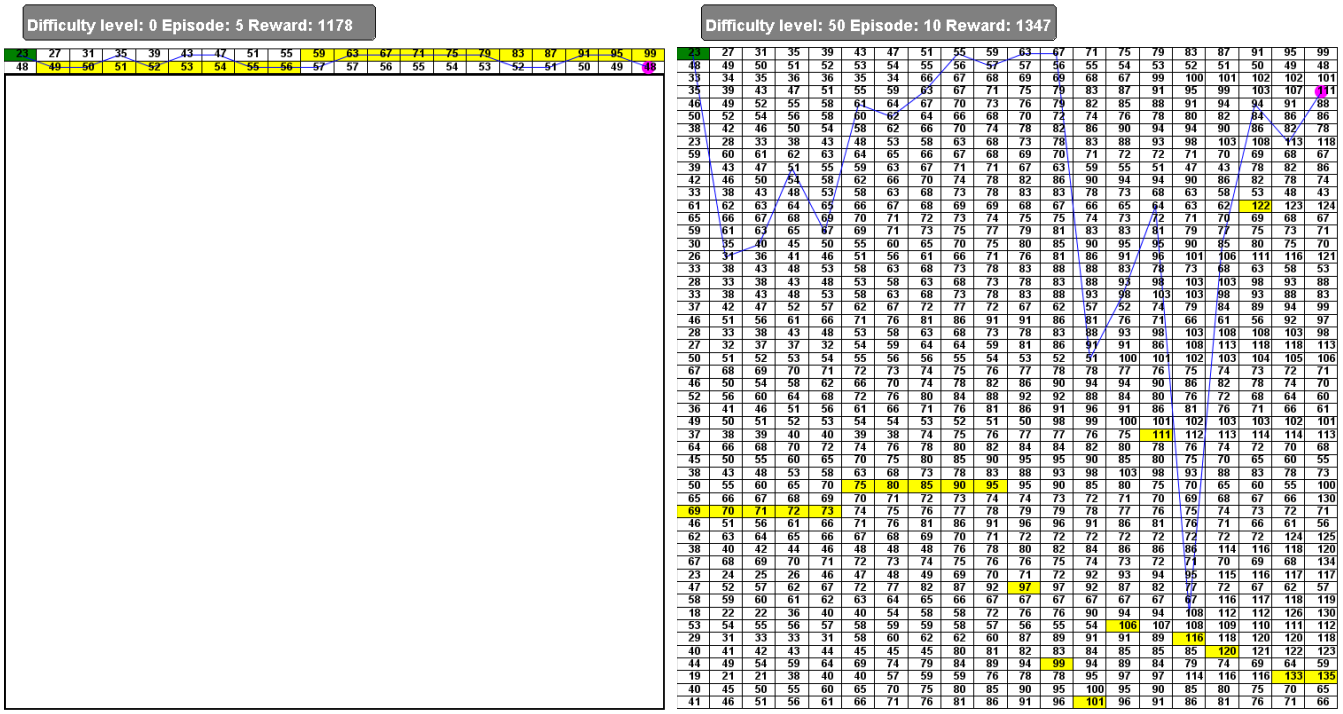
This section elaborates the designs of simulation environments and experimental evaluations conducted to test the proposed Flow-based RL model.

A. SIMULATION ENVIRONMENTS

The experiments utilise two tasks designed to investigate two different objectives:

- A maze navigation task: where the agent is forced to use the new knowledge presented to the system.
- A reward collection task: where the agent is given the option the use the new knowledge presented to the system but is not forced to do so.

Each task consists of 51 challenge levels each presenting new knowledge to the system to evaluate agent performance in traditional RL, incremental RL, and Flow-based RL environments. The first task is associated with a maze navigation environment as depicted in Figure 3. The agent is expected to navigate through the available cells by finding a path avoiding the obstacles (in black) from the start position (red) to the end position (green). The goal is to find the shortest path while avoiding the obstacles. The first challenge involves no obstacles, and the agent has the freedom to explore all cells and find a suitable path to reach the end position (Figure 3a). At each challenge level increment, new obstacles are added by blocking free cells to make the task more complex (Figure 3b). The agent can only travel to its



(a) Challenge Level 0 (b) Challenge Level 50

FIGURE 4. Cell reward collection task. The agent is expected to collect rewards by moving onto 100 cells (figure indicates only 20 cells for clarity of representation) across multiple channels. Switching between channels incur a cost associated with the distance between the two channels. The goal of the agent is to collect the maximum total rewards possible (rewards from cells - costs of channel switching) by the end of 100 time steps. The first task starts with only 2 channels, and each challenge level introduces a new channel with each channel having at least one cell with a higher reward than all previous channels at the same cell position upto 52 channels. The cell with the highest reward for each column is highlighted in yellow. However, this may not be the optimal path since channel switching also incurs a cost associated with the distance.

Von Neumann neighbourhood (the four adjacent cells from the current position), and therefore, to ensure a consistent increase in complexity level, a cell that was included in the previous solution's path (chosen randomly) and it's Von Neumann neighbourhood is blocked as an obstacle at every subsequent challenge level. As such, the incremental and Flow-based RL agents are forced to utilise the new knowledge presented to the environment, as the previous solution they learned would be void due to being blocked by a new obstacle. The traditional RL agent is not impacted by this decision since it does not carry prior-knowledge to the next challenge levels.

The next task that is used for evaluations is illustrated in Figure 4. The ultimate task consists of 52 channels (rows) with the first challenge level starting with only 2 channels. A new channel is added to the task environment at every challenge increment. Each channel consists of 100 cells, and each cell is associated with a reward value. At a given time tick, the agent can move to the next cell of any channel available. There is a cost associated with changing channels which is calculated based on the distance between the current channel and the channel being moved to ($\sqrt{|current channel - new channel|}$). The goal of the agent is to collect the maximum total reward (rewards from cells - costs of changing channels) at the end of 100 steps by identifying the best path through all available channels.

A few design strategies were used to ensure a consistent increase in complexity with the challenge increments. Each channel being added will have at least one cell which has a higher reward than the rewards of all the previous channels at the same cell position. This condition ensures that a difficulty increase is guaranteed with every new channel being added as there is an advantage in moving to the newly introduced channel for a higher reward. The total reward of all cells in a single channel should be within a given range [1000-1500] and the rewards are incremented along the channel in a sinusoidal stepwise format. Figure 5 depicts the nature of reward assignment in the cells within each channel.

This task is different from the maze navigation task where the incremental and Flow-based agents would be forced by the design itself to utilise the new knowledge presented at each difficulty level. In this case, the agent is provided with the choice to either explore the new knowledge or to remain with the solution identified at the previous difficulty level. The new channel added would be useful to explore since it has at least one cell with a higher reward than all cells of the previous channels in that particular column, but the agent is free to decide whether to visit that channel or not.

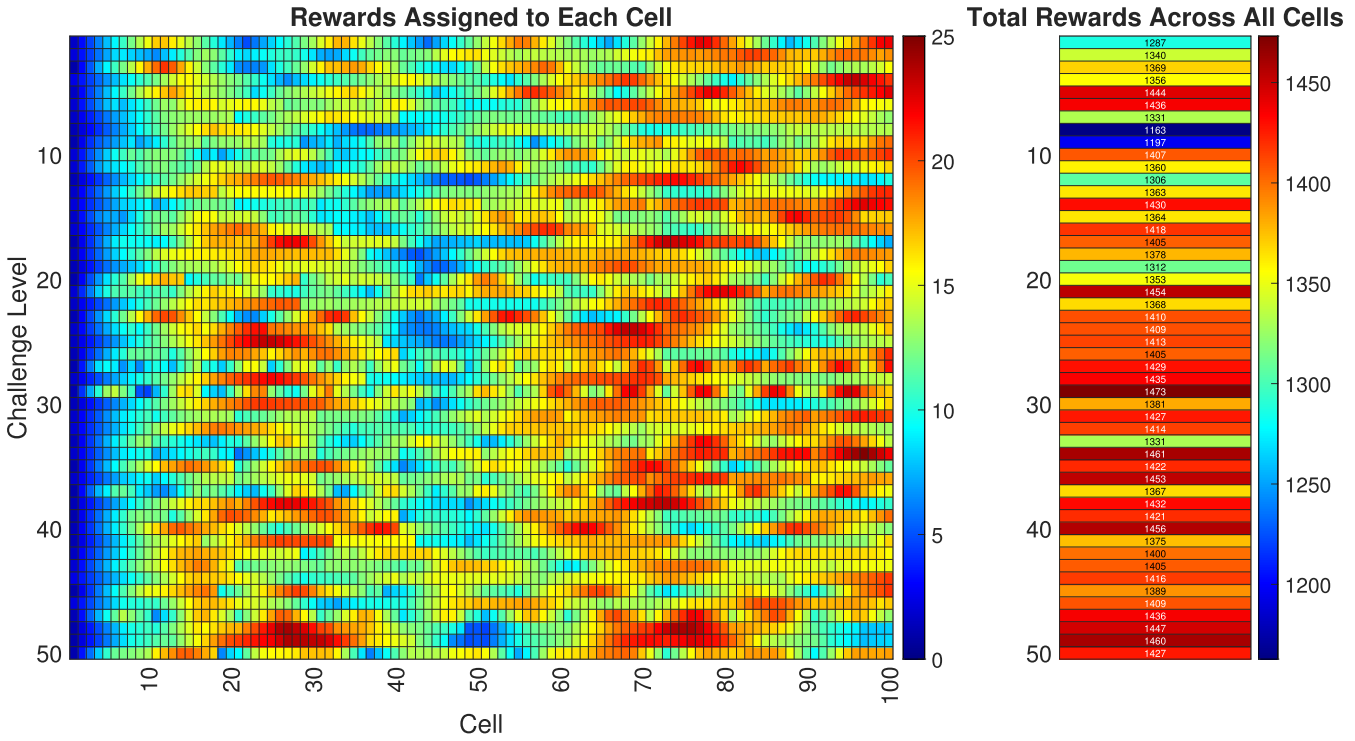


FIGURE 5. Assignment of rewards to cells in the cell reward collection task. The rewards are assigned to each cell in each channel to ensure a consistent increase in the complexity across the introduction of new channels. The rewards across the columns are incremented along the channel in a sinusoidal stepwise format while ensuring the total rewards in a single channel is always within the range [1000-1500]. The figure depicts the rewards assigned to cells averaged across 50 simulation environments.

TABLE 1. Attributes of the setup for flow-based reinforcement learning.

Attribute	Value
Challenge Levels	0-50
Boredom Threshold (φ)	-1
Decay Constant (ε) Initial: for epsilon-greedy q-learning	0.4
Decay Constant (ε) Update	$\varepsilon * 0.9$
Decay Constant (λ_1)	0.9
Decay Constant (λ_2)	0.5

The attributes of the setup for the RL models are as given in Table 1. All 3 models are setup with the same decay constant ε for greedy learning for a fair evaluation.

B. RESULTS

This section evaluates the experimental results obtained by exploring the two task environments described above. The proposed Flow-based model is compared against a traditional RL model and an incremental learning model which were designed with the same common parameters used for the Flow model for a fair comparison. The traditional RL model is run for each challenge level independently to determine the solutions. The incremental learning model can use the accumulated knowledge from previous challenge levels in developing a solution for the new challenge levels. The criteria to update to the next challenge level is defined based on the stabilisation of error (to determine the convergence of the solution) for the incremental learner. All experiments with the 3 models were repeated for 50 different seeds each and the aggregated result was considered during evaluations. Two-sample t-test and one-way ANOVA are adopted as the statistical evaluators for comparisons of two and three groups respectively, and the statistical significance level is set at $p = 0.05$.

Figure 6 demonstrates the analysis results of the maze navigation environment averaged across 50 runs each. The total time taken (the number of state, action pairs visited across all episodes) for each challenge level is shown in Figure 6(a). The agent learning a particular challenge with the Flow-based model or the incremental model builds on the knowledge acquired from previous challenge levels. Therefore, a fair comparison should use the cumulative time of the Flow/incremental model where the aggregated total time taken to learn all challenges up-to each specific challenge is considered. The time curves plotted for both Flow and incremental models in the figure demonstrate this cumulative time. The time curve for the traditional model depicts the time taken to run only through each specific challenge. Accordingly, it can be seen that the Flow-based model takes significantly less time than the traditional model ($p = 3.0924\text{e-}24 < 0.05$) and the incremental learning model ($p = 8.2731\text{e-}04 < 0.05$). Even though the cumulative time is considered for the Flow model, the time taken to learn with the traditional model

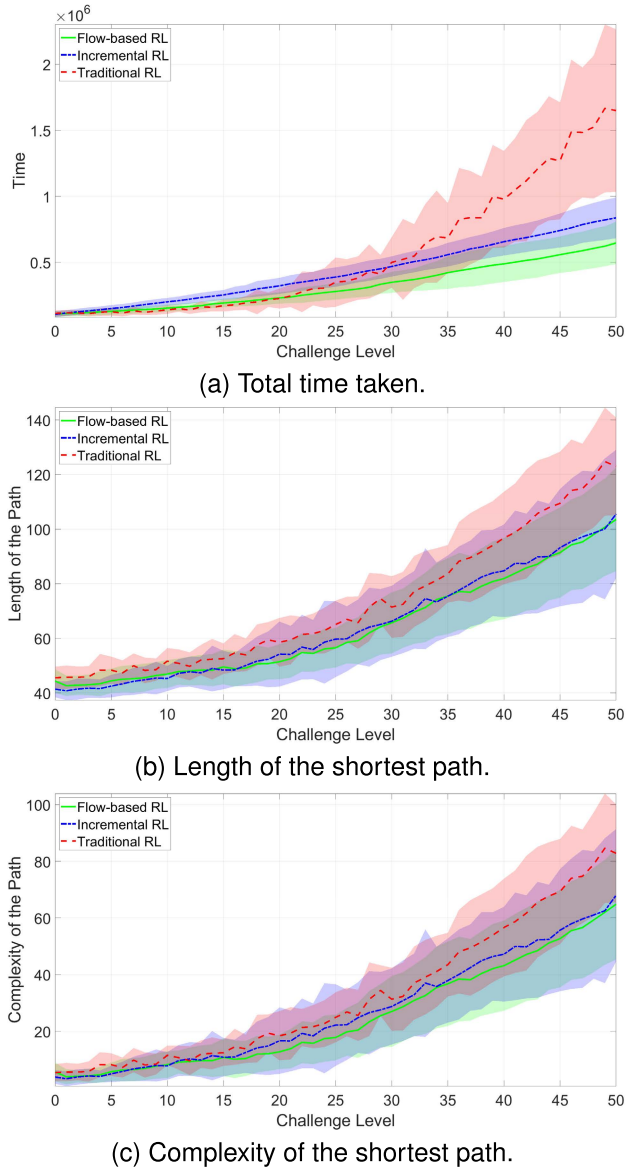


FIGURE 6. Analysis of the maze navigation task environment for each challenge level. The experimental results are averaged across 50 runs each and the shaded areas depict the standard deviation.

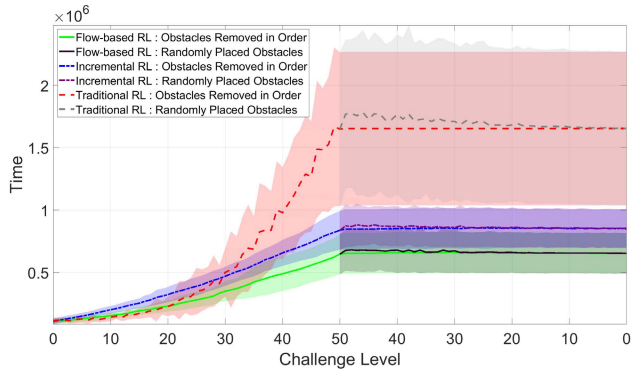
starts increasing significantly than the Flow model after the 20th challenge level. Therefore, as the complexity of the task increases, the traditional model finds the task to be increasingly difficult compared to the Flow-based model. Similarly, the Flow-based model is also consistently more efficient than the incremental model.

In close examination of the time curves, it can also be seen that even the traditional RL model performs significantly faster than the incremental RL model up to the 30th challenge level ($p = 1.3021\text{e-}16 < 0.05$), only after which the incremental model starts performing more efficiently than the traditional model. This can be explained by the design of the task. For traditional RL model, the increasing number of obstacles makes the task difficult with the learner having to learn to avoid all obstacles at every challenge level as it does not carry forward any prior knowledge of the environment. Therefore, the time required consistently increases across challenge levels. However, the incremental learner possesses prior knowledge of the environment from the previous challenge levels and according to the results it can be deduced that it overfits and finds it difficult to adapt to a new path taking more time to converge to a solution until around the 30th level. However, as the complexity of the challenge level further increases, the number of obstacles increases, thus gradually reducing the number of path options to reach the exit (solutions). Therefore, even if the incremental learner is prone to overfit, it becomes relatively easier to identify a new solution as the complexity increases for two reasons: the learner is forced to look for a new path through the design itself (as the new obstacle always intercepts the previously identified solution); and the available options for a solution gradually decreases with the increasing number of obstacles. In contrast, this phenomenon is not observed with the Flow-based learner. The tendency of the incremental learner to overfit is further investigated with the reward collection task.

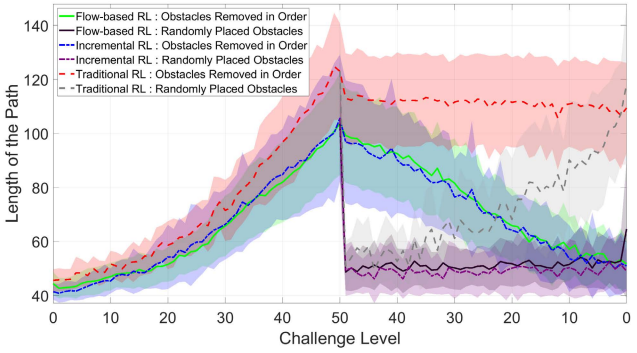
The length of the shortest path identified for each challenge level with increasing number of obstacles is shown in Figure 6(b). The lengths of the paths increase with all 3 models as the challenge level increases due to the increasing number of obstacles that should be avoided to reach the end position. At a glance, the Flow-based model and incremental learning model seem to identify shorter paths for all challenges compared to the traditional model; however, there is not enough statistical evidence to suggest a significant difference between the solutions derived by the models ($p = 0.0615 > 0.05$).

The complexity of the paths were determined based on the Manhattan distance which is the the sum of the absolute differences between the start and the end positions. The difference between the actual path distance and the Manhattan distance was considered as the complexity of the path. According to Figure 6(c), the complexities of the paths identified by all 3 models are increasing with the increasing number of obstacles and the longer paths that should be followed as a result. This observation further supports the design decision of the challenge levels as the increasing complexity of the paths correspond to an increasing complexity of the challenges. However, similar to the path length results, there is no statistically significant difference between the complexities of the paths identified by the models ($p = 0.1817 > 0.05$).

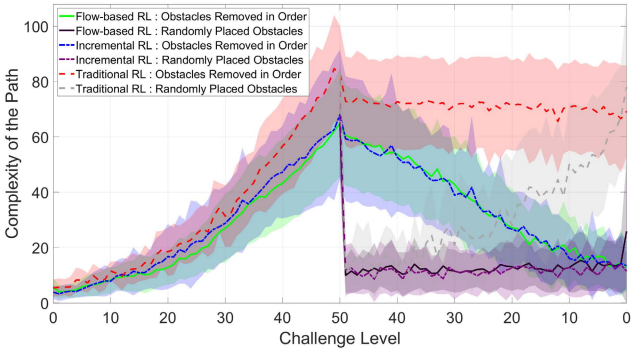
Therefore, the results suggest that the Flow-based model is significantly efficient at identifying solutions for increasingly complex challenges; however, there is not enough evidence to suggest an improved quality of the results compared to the traditional and incremental model with the current observations. In order to further explore the applicability of the Flow-based RL model, the evaluations were then extended to more dynamic scenarios. The next set of experiments were conducted to analyse whether the Flow-based model can effectively utilise the skills learned through performing in the



(a) Total time taken.



(b) Length of the shortest path.



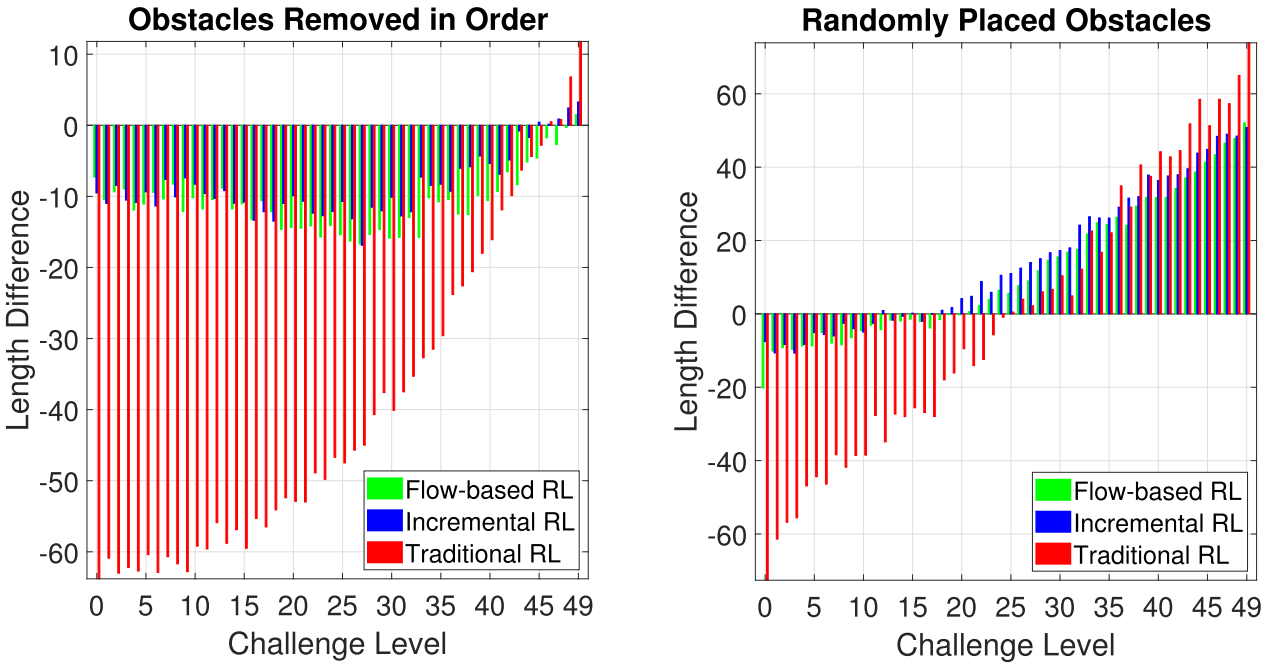
(c) Complexity of the shortest path.

FIGURE 7. Analysis of the maze navigation task environment for dynamic scenarios. The skill level is fixed after the 50th challenge and the agent is then made to navigate in: an environment where the added obstacles are removed in reverse order for each challenge level; and in an environment where a decreasing number of obstacles are placed in random places for each challenge level for 50 more challenges. Evaluations are shown for all 100 challenge levels. The experimental results are averaged across 50 runs each and the shaded areas depict the standard deviation.

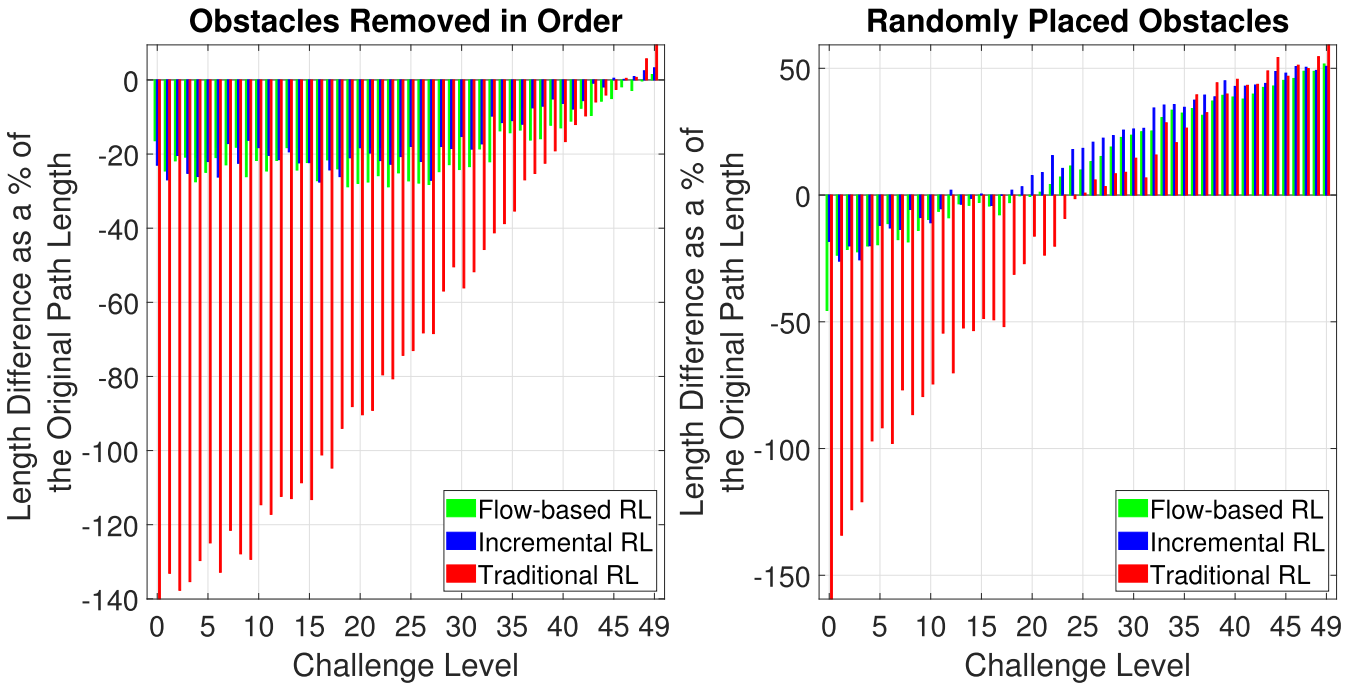
Flow zone when the task environment changes. Two sets of evaluations were conducted in this regard. As the first experiment, the skill level (Q-table values) was fixed after the agent has completed learning the 50th challenge with 50 different obstacles. The evaluations were then conducted by removing each obstacle in the reverse order that was added. Each evaluation after the 50th challenge would then commence with the skills learned upto the 50th level. The same experiment was repeated with the traditional and incremental RL models where the skill level at the 50th challenge was then fed into the agent as the commencing skill level for each decrementing challenge level. For the second experiment, the obstacles were not removed in order; rather, a decreasing number of obstacles were placed at random positions in the environment. I.e., after the agent has completed 50 challenges, the next challenge is to overcome 49 obstacles placed in a different random order to what the agent has experienced so far. The subsequent challenges include decreasing number of obstacles upto no obstacles placed at random positions. The skill level achieved by the end of 50 challenges is fed to the agent for each challenge after the 50th level as before.

Figure 7 illustrates the results for the total of 100 challenge levels for Flow-based, incremental, and traditional RL models as mentioned. According to Figure 7a, all 3 models show that they take a statistically insignificant time to learn the next challenge after the 50th challenge with their already improved skill level with both experiments (obstacles removed in order, and placed at random places). However, the Flow-based model is still capable of completing the task faster than the other two models.

Further, the length of the shortest paths shown in Figure 7b demonstrates that the Flow-based and incremental learning models can significantly enhance the agent's skills towards achieving significant performance levels. When the obstacles are removed in order, the lengths of the paths identified gradually start decreasing implying that the Flow-based and incremental models can use previously learned knowledge to fall back on to a simpler challenge. More importantly, when the obstacles are placed in random places, they behave significantly better and shows the capacity to find relatively similar shorter paths during all challenge levels despite the complexity of the challenge. This observation deduces that the models did not in fact master the skills for a specific challenge but achieved higher performance levels which lead the agent to be able to tackle any dynamic goal in the given problem space. On the other hand, the traditional RL model shows that when obstacles are removed in order, the model is not capable of finding shorter paths any longer even though the challenge is being simplified. The lengths of the paths that are found when the agent is presented with simpler challenges after the 50th challenge is in the same range as the solution derived for the 50th challenge. This demonstrates that the traditional model has overfitted and is incapable of readjusting to different challenge requirements. When the obstacles are placed at random places, the sudden significant drop in the path length can be observed at the 49th level, but it starts performing poorly as the challenge level reduces. This interesting observation is due to the same overfitting issue observed before. Once 49 obstacles are placed in random order, it suggests that a majority of the grid is covered with obstacles which will significantly obstruct the path identified at the 50th level forcing the agent to learn a new path disregarding some of the knowledge gathered earlier. This leads to discovering a shorter path with the higher number of randomly placed obstacles. However, as the number of



(a) Absolute length difference.

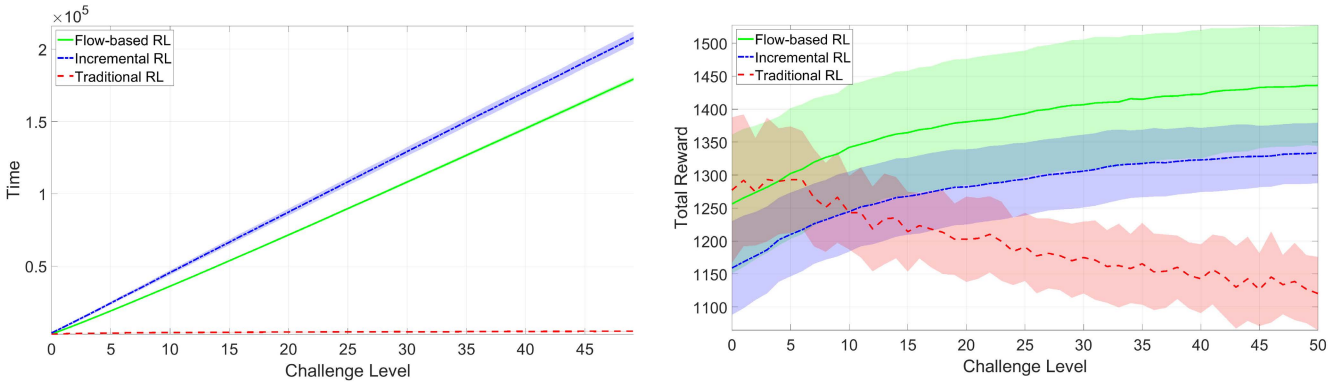


(b) Length difference as a percentage of the length of the original path.

FIGURE 8. Length difference for the shortest paths derived by the three models for each challenge level from 0 leading upto 50th versus challenges after 50th level upto 0th level (original-new length). The experimental results are averaged across 50 runs each.

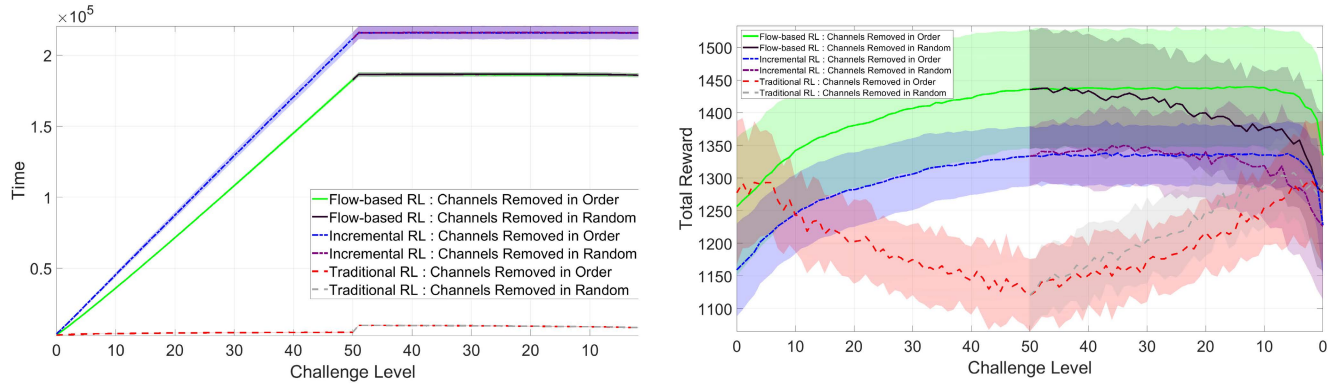
obstacles are reduced, the previous path is impacted less significantly, and as the model does not have generalised knowledge it simply tries following the same old inefficient path despite having shorter alternatives which never get explored.

The results are similar with the complexity of the paths observed in 7c as the complexity is correlated with the length of the path. To further analyse the paths derived by the models from 0-50th challenge versus 50-0th challenge after commencing the learning process from the skill set of 50th



(a) Total time taken. (b) Total rewards collected.

FIGURE 9. **Analysis of the reward collection task environment for each challenge level. The experimental results are averaged across 50 runs each and the shaded areas depict the standard deviation.**



(a) Total time taken. (b) Total rewards collected.

FIGURE 10. **Analysis of the reward collection task environment for dynamic scenarios. The skill level is fixed after the 50th challenge and the agent is then made to collect rewards while minimising costs in: an environment where channels are removed in reverse order for each challenge level; and in an environment where channels are removed from random locations successively. Evaluations are shown for all 100 challenge levels. The experimental results are averaged across 50 runs each and the shaded areas depict the standard deviation.**

level, Figure 8 illustrates the length differences (original-new). When the obstacles are removed in order, this shows that none of the three models are capable of finding a solution as accurate as the original solution except for a few higher level challenges, but traditional model demonstrates the worst performance out of the three. When the obstacles are placed in random locations, the challenge levels with less obstacles than around 20 show inaccurate performances, but starts improving the results for more complex challenges. Similar to the previous observation, the traditional model shows the highest negative length difference. This further supports the evidence to suggest that the traditional RL model is not as robust and flexible in a way that it can utilise its learned skills to overcome a new goal successfully.

The next set of results illustrated in Figure 9 analyses the second problem which is on collecting cell rewards. The previous environment identified that when the agent is forced in the learning process to utilise new knowledge through the task design, both Flow and incremental models can find similar results; however, Flow is more efficient in comparison to incremental learning. The primary goal of investigating the next environment is to understand the behaviour of these models when the agent is not forced but is only given the choice to utilise new knowledge through new difficulty levels of the task.

In contrast to the observations with the maze navigation task, the time analysis presented in Figure 9a depicts that the cumulative time taken by the Flow model increases exponentially with the challenge level and is significantly higher than the traditional model ($p = 0.0 < 0.05$). The incremental learning model is taking even more time compared to the Flow model ($p = 0.0 < 0.05$) and is the most inefficient out of the 3 models compared. On a similar note, unlike the maze navigation task where a statistically significant improvement in performance was not observed, Figure 9b shows that the total rewards (cell rewards - channel switching costs) collected by the Flow-based RL model significantly increases than both the traditional model ($p = 0.0 < 0.05$) as well as the incremental learning model ($p = 0.0 < 0.05$). This shows that as the number of channels increases, the traditional model finds

it increasingly difficult to collect rewards while reducing the cost associated with channel switching. Incremental learning model, on the other hand, does collect higher rewards but cannot reach the performance level of the Flow model. This provides further evidence to support the previous observation with the maze navigation task where the incremental learner is prone to overfit (although less significantly compared to the traditional model) and not investigate new knowledge as comprehensively as the Flow-based learner.

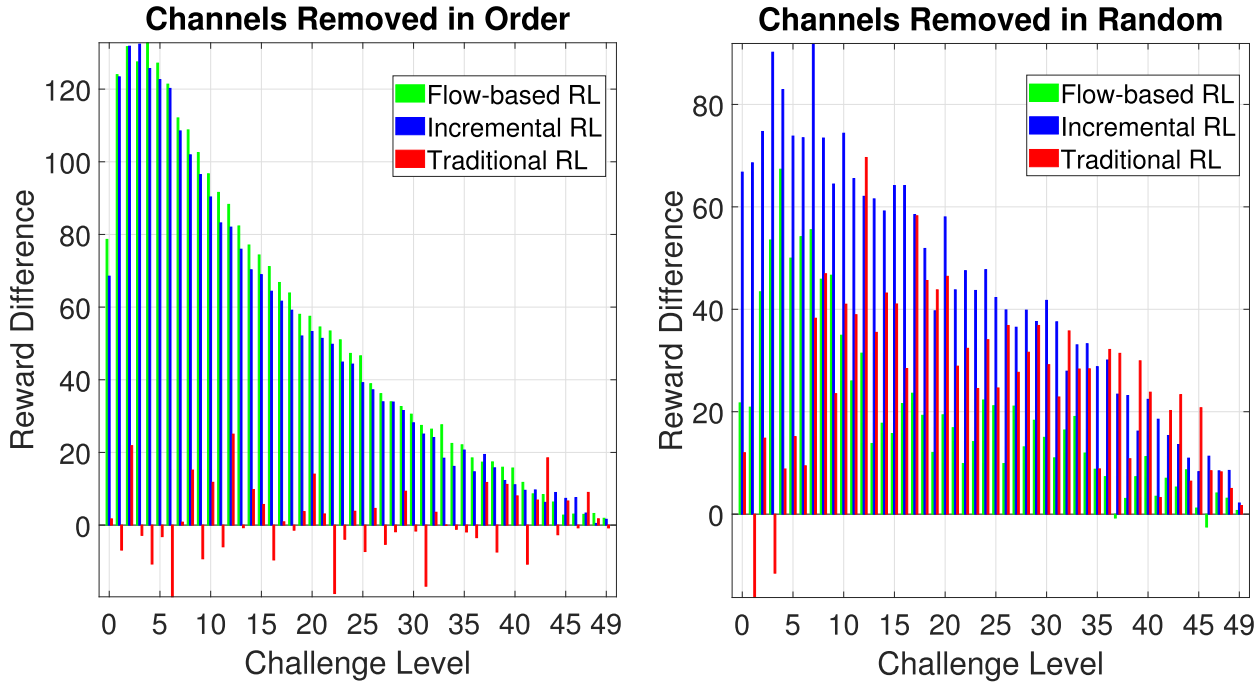
To further analyse the performance of the models during more dynamic and complex environments, Figure 10 illustrates the results when the skill set of the 50^{th} challenge was used as the starting point to learn each challenge where channels are removed in succession in reverse order of introduction, and when channels are removed from random locations. Similar to the observations with the maze navigation task, the time taken to achieve the challenges after the 50^{th} level does not improve significantly for all three models as they have a boost in learning with the already improved skill level fed to the agent.

According to Figure 10b, when the channels are removed in reverse order, the Flow-based model is capable of effectively utilising its learned skills to collect a higher amount of total rewards for all challenge levels. The incremental learning model has the same pattern of reward collection, but consistently collecting lesser rewards compared to the Flow-based model ($p = 0.0 < 0.05$). The traditional model, on the other hand, shows a similar performance to its original results where the total rewards is less for challenges with more channels and gradually increases when the channels are reduced. The reward differences (new reward - original reward) shown in Figure 11 show that the Flow-based model and the incremental model consistently maintain a higher performance compared to their original performance when the channels are being removed after the 50^{th} level. But the traditional model frequently achieves less rewards than the original for certain challenges. When the channels are removed randomly, the performance of the Flow model is not as high as when the channels are removed in order, but still better than the original for all challenges except for 2 according to reward differences in Figure 11. The incremental learning model shows a higher performance level compared to its original learning phase, however, as discussed before, the rewards collected are not as high as the Flow model. The performance of the traditional model also improves than its counterpart which illustrates a similar observation to the maze navigation task. As the model is forced to change its course due to certain channels being not available from the path learned at the 50^{th} level, the performance improves upto a certain level. However, there exists statistically significant evidence ($p = 0.002 < 0.05$) to suggest that both results are not difference from each other for when channels are removed in random versus in order for the traditional model.

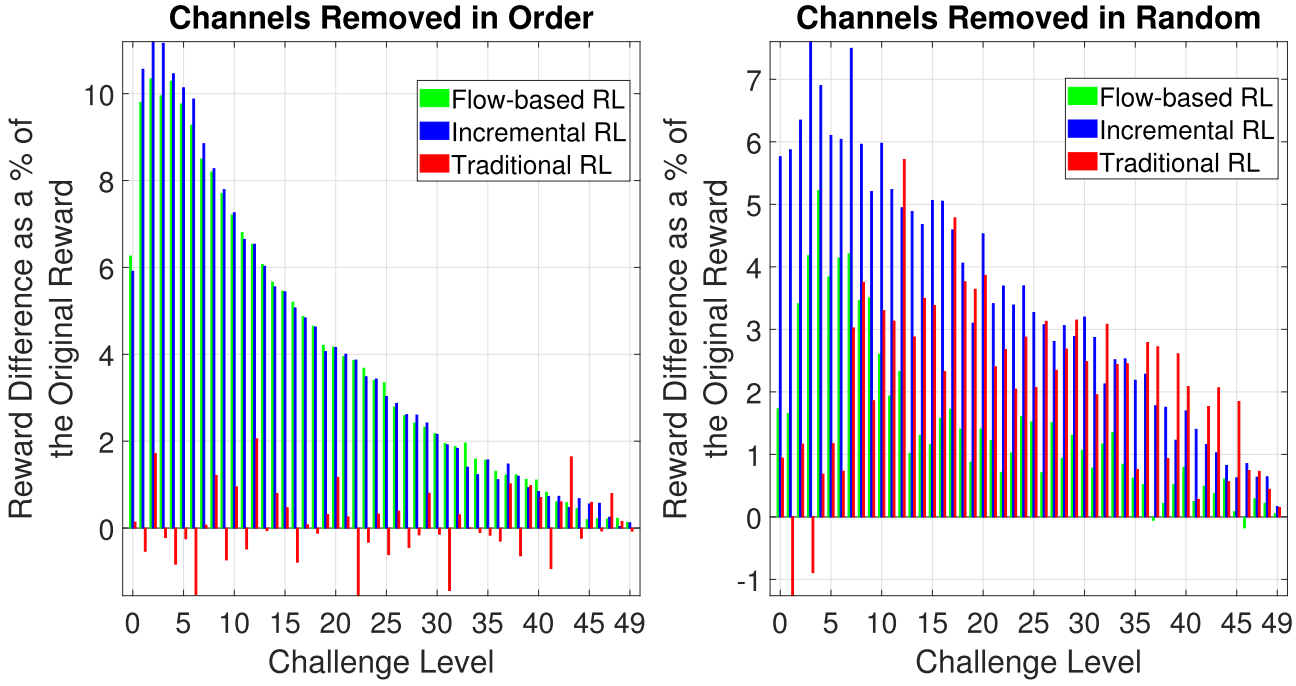
To better understand the causes for the performance of the models as observed, Figure 12 looks at the movement across channels for the agent during all 100 time steps at each challenge level for both models. This gives a clear understanding of the differences observed in the total rewards collected. The traditional model starts with smaller channel switches during the less complex challenges (as expected due to unavailability of a large number of channels). But as more channels appear in the task, the model jumps to these channels for the higher rewards disregarding the costs associated with transferring through channels. When the channels are removed in random, it can be seen that the model switches back and forth from distant channels to collect more rewards, however, it is also associated with higher costs which result in a low value of total reward. Conversely, the Flow model and the incremental learning model behave more intelligently taking the cost into consideration. These models identify that the most efficient pathway is to move within a smaller set of channels and collect the best rewards from them rather than switching to distant channels that can increase the overall cost. This pattern is more significant in the Flow model with even less distant switches being observed compared to the incremental learning model. Despite each higher channel having some cells with better rewards, the model has the ability to compare the relative benefits of exploration versus exploitation to identify the best solution to improve rewards.

This is further analysed with Figures 13, 14, and 15 which evaluate the individual cost and reward values. According to the figures, the performance difference observed with the traditional model is caused due to the cost associated with switching channels. The traditional model consistently scores higher costs during each cell movement due to constantly switching to channels that are further apart. As the challenge complexity increases, the cost values increase proportionally due to the availability of more channels. Conversely, the Flow-based model and the incremental model are capable of maintaining the costs at a minimum by only switching to channels that are only adjacent and focusing on enhancing the reward pool while retaining a minimum cost damage. Despite the same strategy used by both the Flow model and the incremental model, the Flow model is still capable of collecting significantly more rewards than the incremental model as discussed before. The average rewards collected for each challenge level depicted in Figure 13b illustrates this difference in rewards. Therefore, this deduces that the incremental learner is more reluctant to adapt to new knowledge and tends to stick with the knowledge gathered previously. As a result, it misses the opportunity to increase the rewards collected. Conversely, the Flow model is more flexible, and is open to explore the new channel presented at every difficulty level while retaining the previously acquired knowledge to minimise costs and collect more rewards.

The evidence further proves the observations made with the maze navigation task that suggests the Flow-based learning model can push the boundaries of traditional and incremental RL models. It has the potential to utilise the skills learned in simpler challenges to achieve more complex goals in future iterations, and the robustness and flexibility to adapt to dynamic situations. In contrast to the compared models,



(a) Absolute reward difference.



(b) Reward difference as a percentage of the rewards originally collected.

FIGURE 11. Difference in total rewards collected by the three models for each challenge level from 0 leading upto 50th versus challenges after 50th level upto 0th level (new reward - original reward). The experimental results are averaged across 50 runs each.

Flow-based agents improve their skills in line with the challenges while not being bored or exhausted. The evaluation results deduce that such an agent that learns in a Flow zone can not only learn to achieve a goal but learn to enjoy the experience while building awareness of the environment improving robustness and fault tolerance.

V. CONCLUSION AND FUTURE WORK

This paper investigates a novel Flow-based RL model as a potential alternative to overcome the challenges associated with modelling artificial agent systems that can adapt to complex and dynamic environments. The existing AI techniques such as incremental and transfer learning suffer from

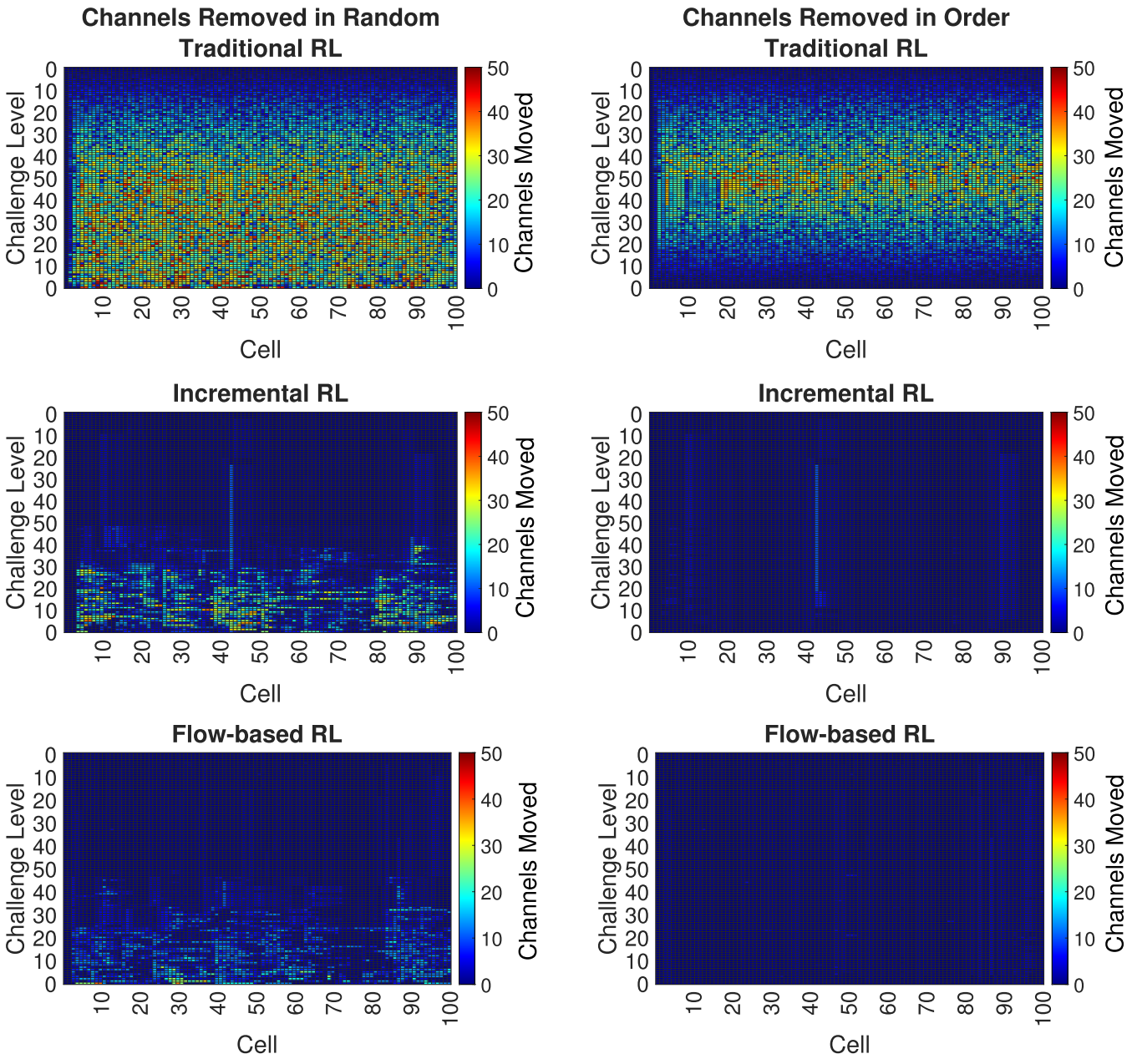
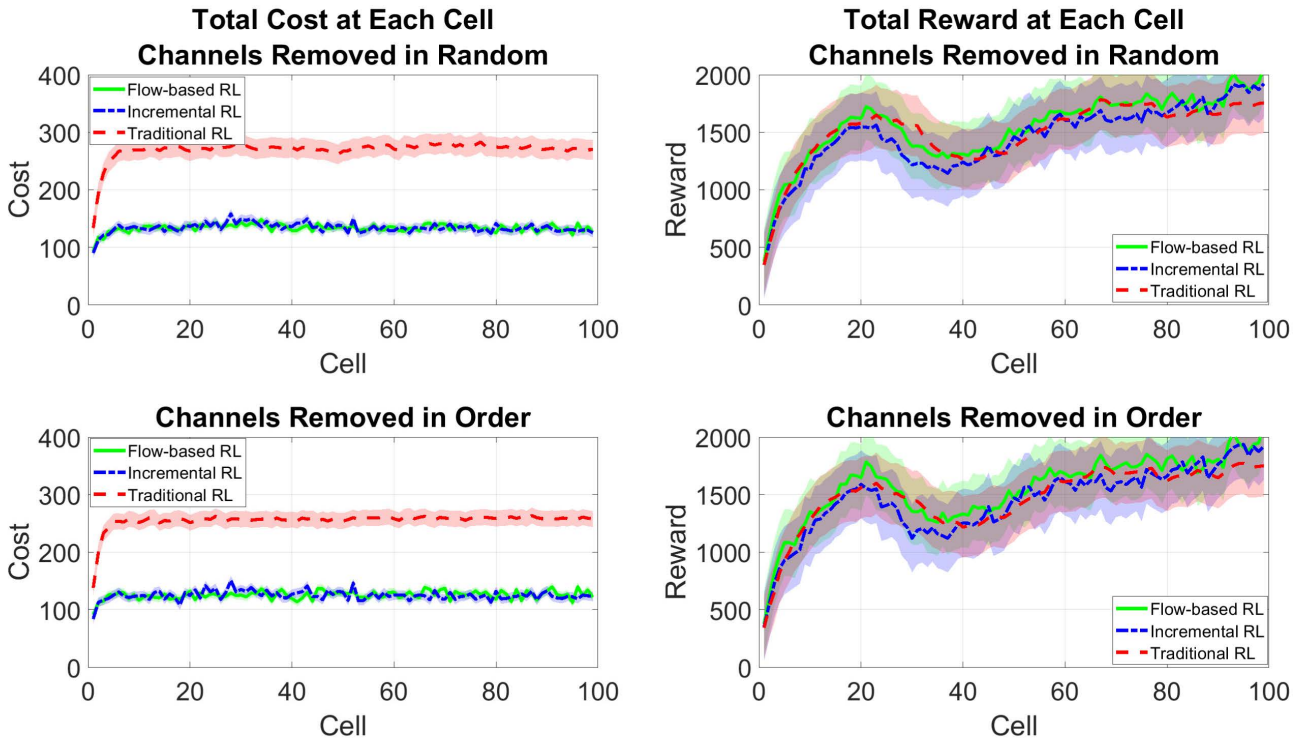


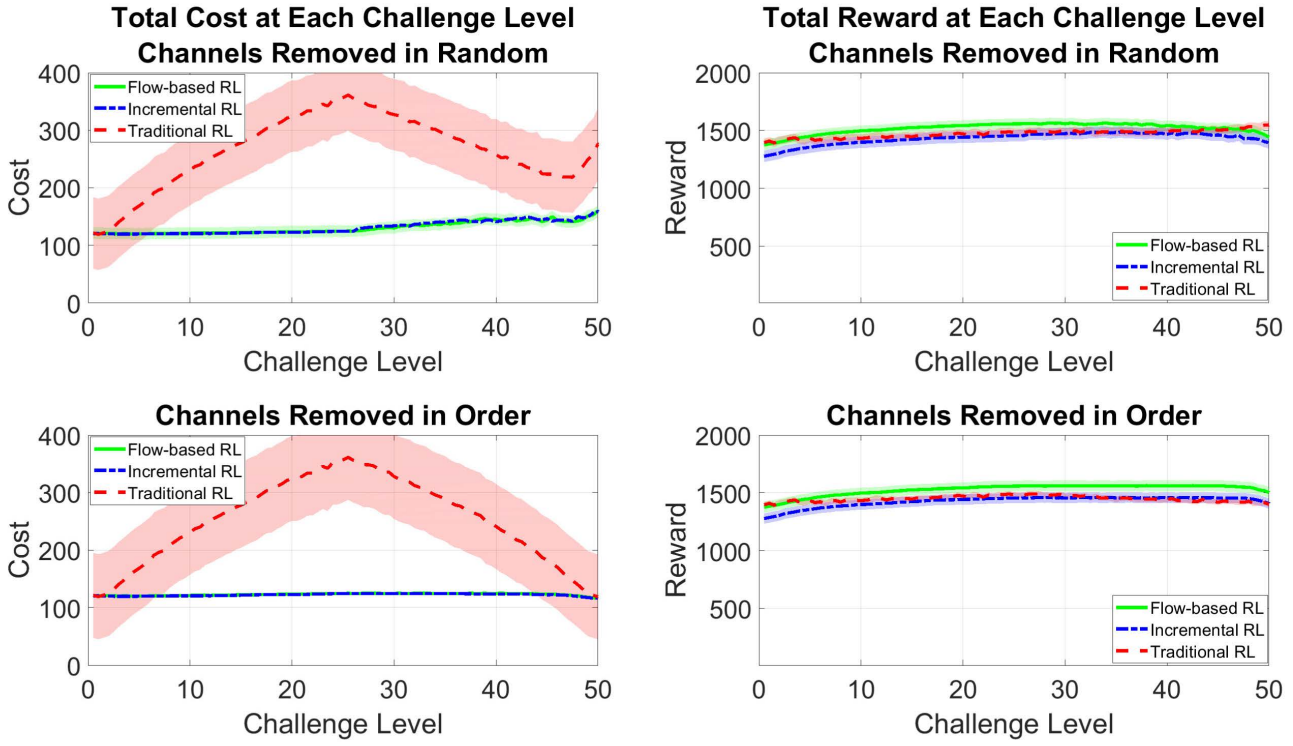
FIGURE 12. Channels moved to by traditional, incremental, and Flow-based RL models at every time step for all 100 challenge levels for both approaches: channels removed in order, and in random. The results are averaged across 50 runs each and the colour-bar depicts the channel number for each cell the agent was in at every time step.

issues related adapting to dynamic environments due to the inherent property of these approaches being primarily goal oriented [15]. As a result, these systems lack the capacity to build an awareness of the environment making them less robust in changing environmental conditions. The model proposed here focuses on maintaining agents in a Flow zone, thus enabling them to enjoy an optimal experience of the task which is not fueled only by the external goals but also by the intrinsic curiosity to improve skills in a given task environment. Therefore, agents learn to achieve goals through incremental complexity levels while adjusting their skills set to face any random variation of the task at every complexity level. A measure of identifying the Flow zone is also introduced based on the novelty of the solutions identified by the agents.

The Flow-based model is tested in two simulation environments: a maze navigation task, and a reward collection task with comparisons against a traditional RL model and an incremental RL model to investigate the impact of the proposed modifications to the algorithm. The two environments were designed such that the maze navigation task deliberately forces the incremental and Flow-based models to investigate



(a) Cost and reward collected at each cell across all challenge levels.



(b) Cost and reward collected for each challenge level across all cells.

FIGURE 13. Total cost and rewards collected at each cell and during each challenge level for the traditional, incremental, and Flow-based RL models. Results are averaged across 50 runs each and the shaded areas depict the standard deviation. For the costs and rewards collected at each cell, the results are summed across all challenge levels, and for the cost and rewards collected at each challenge level, the results are summed across all cells.

new knowledge presented to the environment through the incremental difficulty levels; whereas the reward collection task only provides the option for the agent to investigate the new knowledge based on the capacity of the learner. The results indicate that agents learning in a Flow zone has significant advantages over the traditional and incremental

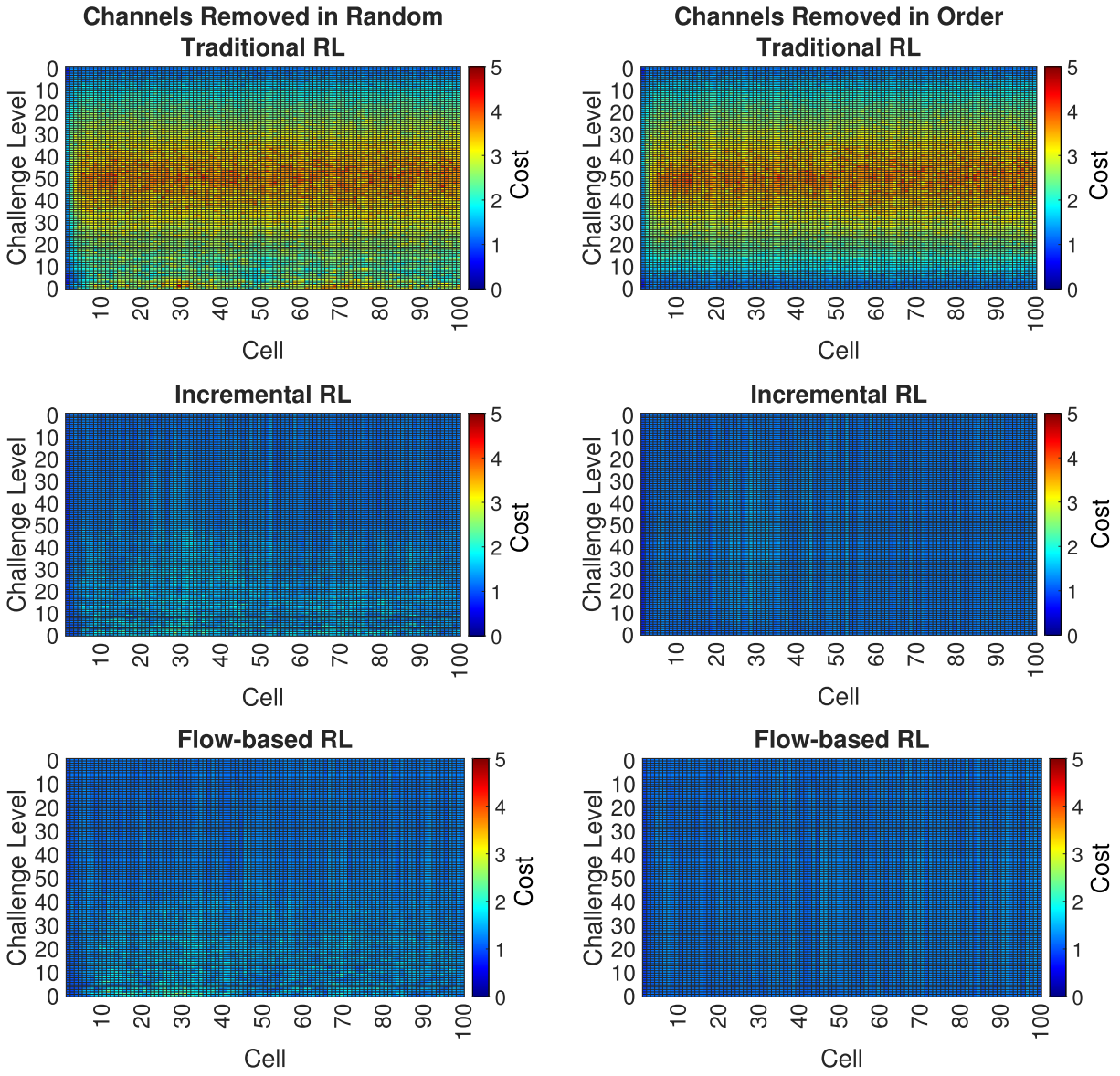


FIGURE 14. Individual cost values collected at each cell by the traditional, incremental, and Flow-based RL models at every time step for all 100 challenge levels for both approaches: channels removed in order, and in random. The results are averaged across 50 runs each and the colour-bars depict the individual cost values collected at every time step.

RL agents. Both simulation results show that the Flow-based agent is able to dynamically adapt to new task environments despite the environmental parameters being different from what they experienced during the learning process. The Flow experience enjoyed by the agent expands the performance level of the agent as it does not choose to transfer to the next challenge level until it can no longer improve its skills in a particular complexity level. Despite already being able to achieve the task, the agent remains in the Flow zone until such time that it is no longer capable of identifying a novel solution leading it to fall out of curiosity to explore and thus move to the next task. An agent trained with the incremental learning model is less efficient compared to the Flow model even in an environment where they are forced to utilise new knowledge presented in incrementing difficulty levels. When the agent is only provided with the choice but is not forced, it is more prone to overfit to the previously learned knowledge and not explore novel solutions which could have lead to better performance. With an agent trained using a traditional Q-learning RL model, the model is driven only by the external goal and is satisfied once it achieves the goal. Therefore, it does not attain enough knowledge to derive flexible solutions given a dynamic environment.

Csikszentmihalyi identifies the contributing factors of Flow as: challenges that match skills; merging of actions and awareness where the attention of the individual is concentrated on the stimuli; clear goals and immediate feedback; making control possible; facilitating concentration and

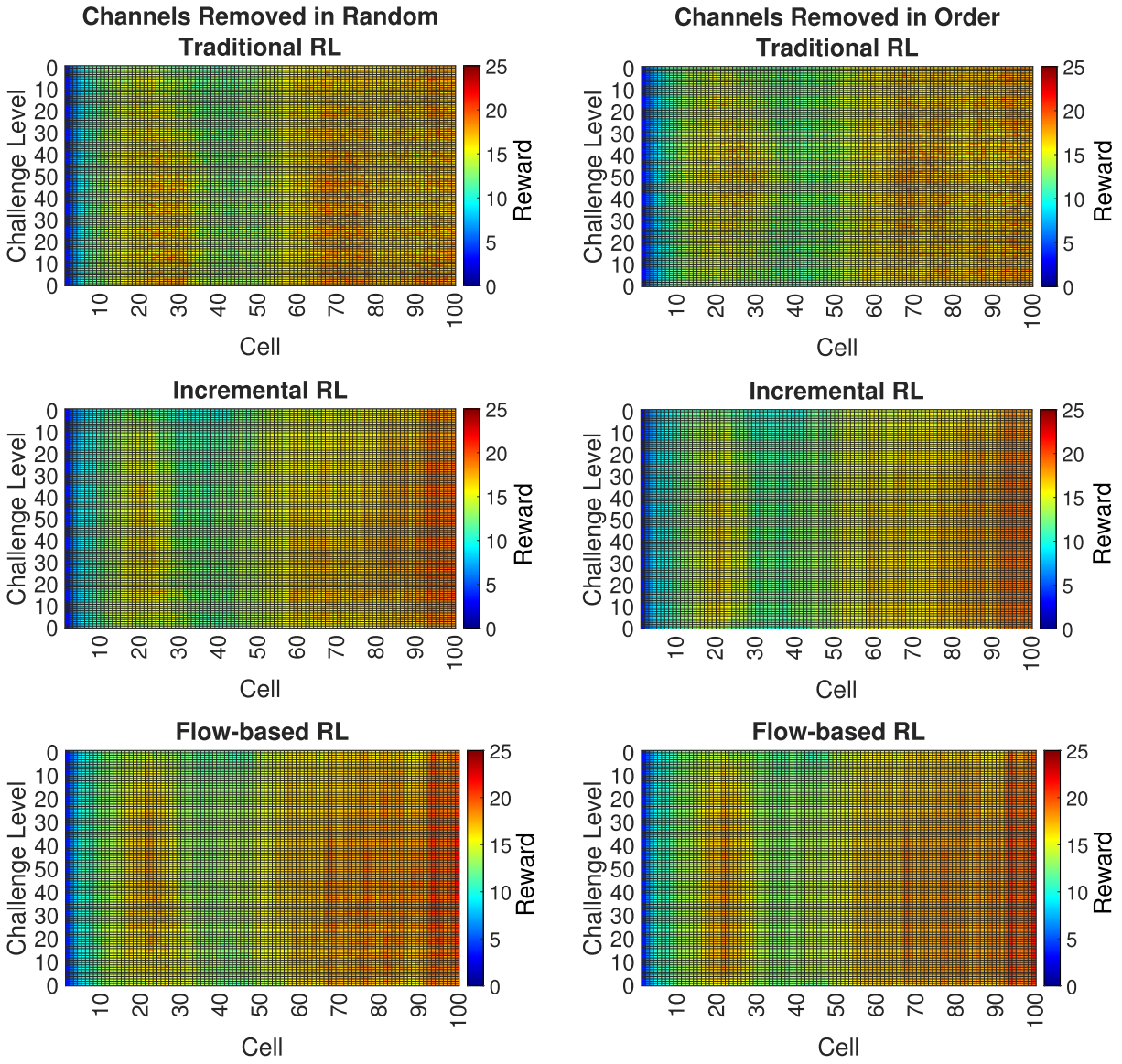


FIGURE 15. Individual reward values collected at each cell by the traditional, incremental, and Flow-based RL models at every time step for all 100 challenge levels for both approaches: channels removed in order, and in random. The results are averaged across 50 runs each and the colour-bars depict the individual reward values collected at every time step.

involvement (loss of self-consciousness); and transformation of time as the agent's own sequences of events marking transitions through states without regard to equal intervals of duration [14]. The proposed Flow-based model follows these factors and demonstrates these characteristics as expected. The task designs were given specific consideration to ensure clear goals and immediate feedback are provided to the agent for the Q-learning process. The two simulation tasks show different time consumptions in comparison to the traditional and incremental models illustrating that the Flow model makes transitions across challenge levels based on its own pace of achieving the optimal experience. The ability to retain performance despite introducing random obstacles and random channel removals for the two environments illustrate that the agent has merged its actions and awareness to facilitate the sense of involvement and control leading to robust performance levels that can achieve dynamic and complex goals.

The evaluations provide promising evidence to explore Flow as a tool to model artificial agents that can perform in complex real-world problem domains with dynamic and constrained environments. This paper investigated Flow in the field of RL, but there exists opportunities to apply the concept with other AI techniques such as artificial neural networks and evolutionary computing. As Flow is a universal concept that is associated with the characteristics of the optimal experience enjoyed by a person/agent, it can be adapted in any AI domain to understand the implications of learning and knowledge transfer during multiple complexity levels of a task. Both simulation environments that are tested within this context are discrete environments where incrementing challenges are relatively intuitive. However, Flow can also

be applied in continuous environments by defining the challenge levels based on agent capabilities. For example, the vision range of an agent can be restricted across multiple challenge levels to increase the difficulty of the task faced by the agent. Therefore, more research directions are also available to investigate the potential of Flow in continuous environments. The proposed Flow-based learning model works with tasks where complexity levels are manually defined. Future research can be directed to explore the possibility to automate the process of defining complexity levels and investigate the performance of the model in environments where complexity cannot be increased in fine improvements. Further, the current results are focussed on single agent systems as the primary concern of this paper is to investigate the applicability of the concept of Flow in AI domains. Therefore, there exists potential to expand the evaluations across multi-agent systems and multi-objective optimisation problems in order to understand the implications when the systems become even more complex. The results presented in this work lend valuable insights into generating agent systems that are equipped with the skills to address more real-world applications.

REFERENCES

[1] I.-J. Liu, U. Jain, R. A. Yeh, and A. Schwing, "Cooperative exploration for multi-agent deep reinforcement learning," in *Proc. 38th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 139, M. Meila and T. Zhang, Eds. PMLR, Jul. 2021, pp. 6826–6836. [Online]. Available: https://proceedings.mlr.press/v139/liu21j.html

[2] M. Dimakopoulou and B. Van Roy, "Coordinated exploration in concurrent reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 80, J. Dy and A. Krause, Eds. Jul. 2018, pp. 1271–1279. [Online]. Available: https://proceedings.mlr.press/v80/dimakopoulou18a.html

[3] Z. Cui and Y. Wang, "UAV path planning based on multi-layer reinforcement learning technique," *IEEE Access*, vol. 9, pp. 59486–59497, 2021.

[4] C. Qu, W. Gai, M. Zhong, and J. Zhang, "A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106099. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620300399

[5] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6252–6259.

[6] Z. Cao, Q. Xiao, and M. Zhou, "Distributed fusion-based policy search for fast robot locomotion learning," *IEEE Comput. Intell. Mag.* , vol. 14, no. 3, pp. 19–28, Aug. 2019.

[7] A. Heuillet, F. Couthouis, and N. Diaz-Rodriguez, "Collective explainable AI: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with Shapley values," *IEEE Comput. Intell. Mag.*, vol. 17, no. 1, pp. 59–71, Feb. 2022.

[8] S. Ju, G. Zhou, M. Abdelshiheed, T. Barnes, and M. Chi, "Evaluating critical reinforcement learning framework in the field," in *Artificial Intelligence in Education*, I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, and V. Dimitrova, Eds. Cham, Switzerland: Springer, 2021, pp. 215–227.

[9] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2156–2162.

[10] M. Riedmiller, A. Merke, D. Meier, A. Hoffmann, A. Sinner, O. Thate, and R. Ehrmann, "Karlsruhe brainstormers—A reinforcement learning approach to robotic soccer," in *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone, T. Balch, and G. Kraetzschmar, Eds. Berlin, Germany: Springer, 2001, pp. 367–372.

[11] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?" *Artif. Intell.*, vol. 171, no. 7, pp. 365–377, May 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370207000495

[12] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, Nov. 2005.

[13] O. Buffet, A. Dutech, and F. Charpillet, "Incremental reinforcement learning for designing multi-agent systems," in *Proc. 5th Int. Conf. Auto. Agents (AGENTS)*, 2001, pp. 31–32.

[14] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper & Row, 1990, vol. 1990.

[15] A. Cahill, "Catastrophic forgetting in reinforcement-learning environments," M.S. thesis, Dept. Comput. Sci., University of Otago, Dunedin, New Zealand, 2011.

[16] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–25, Jul. 2021, doi: 10.1145/3459991.

[17] P. Barros, A. Tanevska, and A. Sciutti, "Learning from learners: Adapting reinforcement learning agents to be competitive in a card game," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 2716–2723.

[18] D. Samarasinghe, M. Barlow, E. Lakshika, and K. Kasmarik, *Task Allocation in Multi-Agent Systems With Grammar-Based Evolution*. New York, NY, USA, 2021, pp. 175–182, doi: 10.1145/3472306.3478337.

[19] U. Wilensky and W. Rand, *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems With NetLogo*. Cambridge, MA, USA: MIT Press, 2015.

[20] P. Maes, "Modeling adaptive autonomous agents," *Artif. Life*, vol. 1, no. 1_2, pp. 135–162, Oct. 1993.

[21] J. L. Elman, "Incremental learning, or the importance of starting small," Center Res. Lang., Univ. California, San Diego, CA, USA, Tech. Rep. CRL-TR-9101, 1991.

[22] J. H. A. Ng and R. P. A. Petrick, "Incremental learning of planning actions in model-based reinforcement learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3195–3201.

[23] A. Dutech, O. Buffet, and F. Charpillet, "Multi-agent systems by incremental gradient reinforcement learning," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 17, no. 1, 2001, pp. 833–838.

[24] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "A novel incremental learning scheme for reinforcement learning in dynamic environments," in *Proc. 12th World Congr. Intell. Control Autom. (WCICA)*, Jun. 2016, pp. 2426–2431.

[25] Z. Wang, C. Chen, and D. Dong, "Lifelong incremental reinforcement learning with online Bayesian inference," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 1–14, Feb. 2021.

[26] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, no. 7, pp. 1–53, Jul. 2009.

[27] Y. Hou, Y.-S. Ong, L. Feng, and J. M. Zurada, "An evolutionary transfer reinforcement learning framework for multiagent systems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 601–615, Aug. 2017.

[28] H. Plisnier, D. Steckelmacher, and A. Nowé, "Self-transfer reinforcement learning for continuous control tasks," in *Proc. Adapt. Learn. Agents Workshop at AAMAS (ALA)*, 2021, pp. 1–7.

[29] Z. Cao, M. Kwon, and D. Sadigh, "Transfer reinforcement learning across homotopy classes," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2706–2713, Apr. 2021.

[30] H. He and X. Zhong, "Learning without external reward [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 48–54, Aug. 2018.

[31] D. Samarasinghe, M. Barlow, E. Lakshika, and K. Kasmarik, "Exploiting abstractions for grammar-based learning of complex multi-agent behaviours," *Int. J. Intell. Syst.*, vol. 36, no. 11, pp. 6273–6311, Nov. 2021.

[32] I. Sarantopoulos, M. Kiatos, Z. Doulgeri, and S. Malassiotis, "Total singulation with modular reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 4117–4124, Apr. 2021.

[33] Y. Wang, H. He, and C. Sun, "Learning to navigate through complex dynamic environment with modular deep reinforcement learning," *IEEE Trans. Games*, vol. 10, no. 4, pp. 400–412, Dec. 2018.

[34] E. Uchibe, M. Asada, and K. Hosoda, "Behavior coordination for a mobile robot using modular reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Nov. 1996, pp. 1329–1336.

[35] C. Atkinson, B. McCane, L. Szymanski, and A. Robins, "Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting," *Neurocomputing*, vol. 428, pp. 291–307, Mar. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220318439

[36] M. Csikszentmihalyi, *Applications of Flow in Human Development and Education*. Dordrecht, The Netherlands: Springer, 2014.
[37] W. Burleson, "Developing creativity, motivation, and self-actualization with learning systems," *Int. J. Hum.-Comput. Stud.*, vol. 63, nos. 4–5, pp. 436–451, Oct. 2005.
[38] I. I. Bittencourt, S. Isotani, V. Wanick, and A. Ranchhod, "Flow experience in learning: When gamification meets artificial intelligence in education," in *Artificial Intelligence in Education: AIED*, vol. 10948, C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, and B. du Boulay, Eds. Cham, Switzerland: Springer, Jun. 2018, pp. 541–543.
[39] B. Cowley, D. Charles, M. Black, and R. Hickey, "Toward an understanding of flow in video games," *Comput. Entertainment*, vol. 6, no. 2, pp. 1–27, Jul. 2008.
[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[41] C.-X. Lu, Z.-Y. Sun, Z.-Z. Shi, and B.-X. Cao, "Using emotions as intrinsic motivation to accelerate classic reinforcement learning," in *Proc. Int. Conf. Inf. Syst. Artif. Intell. (ISAI)*, Jun. 2016, pp. 332–337.

DILINI SAMARASINGHE received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 2021. She is currently a Postdoctoral Research Associate with the School of Engineering and Information Technology, University of New South Wales, Canberra, Australia. Her research interests include artificial intelligence, autonomous agent systems, machine learning, and serious games.

MICHAEL BARLOW received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 1991. He joined as a Postdoctoral Researcher at the University of Queensland, Australia. He was at the Nippon Telegraph and Telephone's Human Communication Laboratories, Japan. In 1996, he joined at the University of New South Wales, Canberra, where he is an Associate Professor and the Head of the School of Engineering and IT (acting). His research interests include simulation, virtual environments, machine learning, serious games, and human–computer interaction.

ERANDI LAKSHIKA received the B.Sc. degree (Hons.) in computer science from the University of Colombo, Sri Lanka, and the Ph.D. degree in computer science from the University of New South Wales, Canberra (UNSW Canberra), in 2014. In 2009, she joined as an Assistant Lecturer at the School of Computing, University of Colombo. She is a Lecturer with UNSW Canberra. Her research interests include human–computer interfaces, multi-agent systems, computational intelligence, multi-objective optimisation, serious games, and games for health.