8 APPENDIX

Here in sections 8.1-8.7 we suggest further research and improvements to the paper, which might provide more stable results, provide useful insights, and possibly lead to a new branch of research with some connected ideas. In the last subsections 8.8 and 8.9 we elaborate and the choices of the experiment setups.

8.1 MORE ELABORATE DISTRIBUTION OF HYPERPARAMETERS

To improve comparison curves, and the precision of all of the experiments, a higher density of the alpha values and number of forward steps could be added. To find more accurate optimal values, one could use e.g. binary search, as we can see from the comparison graphs with broken lines that they are roughly sorted. More fine-grained distribution would not only give optimal values closer to true ones but might also show certain hidden dependencies (possibly even approximate functions/scales) of the considered hyperparameters on the graphs. We could also set the precision of the density, by, for instance, making a step (e.g. 0.1) from 0 to 1.

8.2 MORE MOVEMENT OPTIONS

The task could also be more elaborate. For instance, diagonal moves (i.e. top-right, top-left, down-right, down-left) could be added, which would require more computational power and possibly more detailed analysis. Still, it is a great extension, as it would show more similar to human-like movement in the real world.

8.3 MORE SOPHISTICATED ENVIRONMENT

Besides the agent's actions, many extensions could be added to the environment. Besides it being on a bigger grid, some obstacles, except cliffs, could be added. Simple examples include walls, but some more elaborate ones would make an environment more interesting and possibly more natural to the real world. For instance, hills/valleys on which the agent moves faster/slower (a step of more or fewer cells).

8.4 RESEARCH INCLUDING OTHER ENVIRONMENTS AND OTHER HYPERPARAMETERS

The paper could be extended to compare best n and α values on other environments: either similar grids or completely different setups resembling similar problems. Besides that, many characteristics of the experiments could be controlled: epsilon values (or general parameter of exploration), number of timesteps, episodes; and besides that, other action policies could be considered.

8.5 FURTHER RESEARCH ON WINDY ENVIRONMENT

Even when including the same environment without any visual changes, more specifics could be added to the wind environment from section 4. Firstly, the wind could be in different directions (possibly in the diagonals, especially including the extension in Appendix 8.2). Secondly, the probability of wind could vary, and the research on that would give great insights into the dependency of performance of the algorithm depending on stochasticity of the environment. Lastly, the strength of the wind could be different (i.e. it could lead to a difference of not 1 cell, but more), providing stochasticity to some extent too. These 3 characteristics could be combined with each other, forming a solid paper bringing the results in stochastic problems.

8.6 PLOTTING OPTIMAL POLICY

Apart from the learning curves of each agent with each setup, plots can also include a line of optimal theoretical policy. However, because of the fact that there are 2 possible starting positions, this would imply division of the runs starting in one position and the other, because they have 2 different optimal paths. However, this removes this stochastic element of the setup, simplifying the problem to some significant extent. A solution to this would be to choose 2 (or more) equal to each other start cells. Meaning, that they would have same optimal values both for deterministic and stochastic

environments.
Let us also note that this theoretical best curve is a straight line, because there is a single (or a couple of equivalent) optimal paths on the map whether with or without the wind.
This line would visualize the true performances of each of the algorithms and hyperparameter values because it would show how far our curves are from the desired performance. Afterall, if the true best curve is much higher on the plot, then both of the algorithms perform very poorly and there is no significant advantage of choosing one over another. This is, gladly, not the case in our experiments, because all of the retuns are negative, and curves on our graphs tend to leap close enough to 0 (and thus even closer to the optimal values). Nonetheless, plotting the theoretical optimal return could provide additional comparison information.

8.7 Alternative comparison

In the paper, we have compared the curves based on their AUC. Here we propose a couple of different measures to compare considered algorithms. The first one is Root Squared Mean Error (RMSE), which however compares obtained performance to the optimal value, and thus would require adjustment noted in the previous section about equivalent starting positions. This type of comparison measure is also used by Sutton & Barto (2018). The second proposal is the average return of the final episode (or a couple of last episodes). This measure focuses more on the maximum obtained return to the end, disregarding the speed of learning at the earlier episodes. Thus, whether this is the value of interest depends on the problem itself, and we think that this is a relevant measure in many setups.

8.8 Alternative ϵ-greedy policy

As behaviour policy, we had $\epsilon - greedy$ policy with the following formula:

$$\pi_{\epsilon-greedy}(a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A|}, & \text{if } a = \arg\max_{b \in A} Q(b) \\ \frac{\epsilon}{|A|}, & \text{otherwise} \end{cases} \tag{8}$$

, which distributes probability of ϵ among all actions, and gives additional $1 - \epsilon$ probability to the best action. This way, the best action so far has a higher probability of being chosen, compared to all other actions (because $1 - \epsilon \geq 0$), which is mathematically more appropriate and intuitive than the alternative version, used in some reinforcement learning researches:

$$\pi_{\epsilon-greedy-prev}(a) = \begin{cases} 1 - \epsilon, & \text{if } a = \arg\max_{b \in A} Q(b) \\ \frac{\epsilon}{|A|-1}, & \text{otherwise} \end{cases} \tag{9}$$

Using this formula, it is not guaranteed that the best action is more probable to be chosen than others: for instance, when $\epsilon = 0.95$ (and $|A| = 4$, as in the considered task): $1 - 0.8 = 0.20 < 0.27 \approx \frac{0.8}{3}$. Even though this is not a practically useful epsilon value, this property shows that the first version of the policy is more mathematically strict.

8.9 Choice of design of the experiment

In this section, we would like to elaborate on the choice of the hyperparameters and the curves in the experiment, and in particular, on the alpha values for the learning curves. Initially, we found optimal alphas for each of the agents (Q-learning, SARSA and Expected SARSA), and were using the same α for each agent separately throughout all of the learning curves. However, we can consider each n-step agent as a separate algorithm, as it has many differences in practical behaviour/calculations. Thus, in the end, we chose best alpha for each of the learning curves